

# **Plazic, Arduino-pohjaisen ohjelmoitavan logiikan vertailu kaupallisiin ohjausjärjestelmiin**

Pasi Kuusela

Opinnäytetyö

Toukokuu 2019

Tekniikan ala

Insinööri (AMK), sähkö- ja automaatiotekniikan tutkinto-ohjelma

Tekijä(t) Kuusela, Pasi	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2019
	Sivumäärä 40	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi <b>Plazic, Arduino-pohjaisen logiikan vertailu kaupallisiin ohjausjärjestelmiin</b>		
Tutkinto-ohjelma Insinööri (AMK), sähkö- ja automaatiotekniikan tutkinto-ohjelma		
Työn ohjaaja(t) Teppo Flyktman, Juho Riekkinen		
Toimeksiantaja(t)		
<p>Tiivistelmä</p> <p>Arduino-mikrokontrolleri on pitkään ollut elektroniikka- ja ohjelmointiharrastajien ja opiskelijoiden käytössä erilaisissa projekteissa. Ohjelmistot ovat pääasiassa olleet vain kyseiseen projektiin tehtyjä räätälöityjä ohjelmia. Markkinoille on tullut teollisuuskäyttöön tarkoitettuja ohjelmoitavia logiikoita, jotka perustuvat Arduinon käyttämiin mikrokontrollereihin. Arduino-ympäristöön on tehty yleiskäyttöinen Plazic-niminen ohjelmisto, jolla voi toteuttaa ohjelmoitavan logiikan ohjaustoiminnot sisältäen myös analogiasignaalin käsittelyn ja säädintoiminnot. Plazicin sovellusohjelmointikieli on lähinnä toimilohko-ohjelmointia. Ohjelmisto on tarkoitettu kevyempiin kohteisiin, ja kaikki operointi tapahtuu sarjaliikennepääteohjelmalla.</p> <p>Opinnäytetyössä verrattiin aiemmin toteutettujen projektien avulla muutamaa muuhun ohjausjärjestelmään tapaustutkimustyyppisesti. Tavoitteena oli saada tietoa ohjelmiston teknisestä soveltuvuudesta, käyttäjäystävällisyydestä, hinnasta ja muunneltavuudesta.</p> <p>Toteutus tapahtui käyttämällä tutkimusaineistona toteutettujen projektien raportteja. Vertailun pohjana oli eri järjestelmien erilaiset tekniset toteutukset. Arvio tehtiin vertailemalla eri järjestelmiä kolmessa erilaisessa käytännön toteutuksessa.</p> <p>Tulokseksi saatiin arvio Plazic-sovelluksen käyttökelpoisuudesta. Sovellus on parhaimmillaan pienehköissä kohteissa. Pahimpina puutteina tutkimuksissa nousi esiin graafisen käyttäjäliittymän puute ja ulkopuolisten väyläratkaisujen puute. Kummatkin puutteet ovat teknisesti toteutettavissa avoimessa lähdekoodissa ja ne ovatkin nousemassa ohjelmiston jatkokehityksessä seuraaviksi kehityskohteiksi.</p>		
Avainsanat ( <a href="#">asiasanat</a> ) Arduino, ohjelmoitava logiikka, PLC, Plazic		
Muut tiedot		

Author(s) Kuusela, Pasi	Type of publication Bachelor's thesis	Date 12.05.2019
		Language of publication: finnish
	Number of pages 40	Permission for web publication: x
Title of publication <b>Plazic, comparing Arduino-based logic to commerical control systems</b>		
Degree programme Electrical and Automation Engineering		
Supervisor(s) Flyktman Teppo and Riekkinen Juho		
Assigned by		
<p>Abstract</p> <p>Arduino microcontroller has been used for long time with electronic and program enthusiast in their projects. The softwares have mostly been used only for certain projects. There have been programmed logic controls based on Arduino appearing to the market. There has been made a universal software to the Arduino environment named Plazic. It is used for PLC-controlling and it can be used to analog signals and PI-control, too. Programming resembles the function block diagram program. Plazic is at its best when controlling small targets. All the operating is done by using the serial monitor program.</p> <p>The software Plazic has been compared to a few other controlling systems. Case study – style verifying has been based on three different project. The goal has been to get information of technical suitability, user friendliness, price and transformability.</p> <p>The executing has been made by using research material from projects made before. Different system customs were at the base of verifying. Evaluating was made by comparing various systems from three different practical executions.</p> <p>The result was an estimation of suitability of the program Plazic. The program is most suitable in small targets. The weakness of the program was lack of graphics screen and interface busses. The both weaknesses can be repaired and they will be the next improvement targets.</p>		
Keywords/tags ( <a href="#">subjects</a> )		
Arduino, programmable logic, PLC, Plazic		
Miscellaneous		

## Sisältö

<b>1</b>	<b>Plazic-ohjelma automaatiokäytössä .....</b>	<b>4</b>
1.1	Plazic-ohjelmisto .....	4
1.2	Tavoite .....	4
1.3	Tutkimusaineisto ja menetelmät.....	4
<b>2</b>	<b>Arduino-pohjainen automaatio .....</b>	<b>5</b>
2.1	Mikroprosessoripohjainen automaatio.....	5
2.2	I/O-liitännät .....	7
2.2.1	Digitaalitulo.....	7
2.2.2	Digitaalilähtö.....	8
2.2.3	Analogiatulo.....	8
2.2.4	Analogialähtö.....	9
2.2.5	Muut liitännät.....	10
2.2.6	Arduino-kokoonpanon toteutus.....	11
2.3	Mikroprosessorin ohjelmisto .....	11
2.4	Ohjelmoitava logiikka .....	12
2.4.1	Logiikan rakenne.....	12
2.4.2	Plazic-ohjelman toiminta.....	14
2.4.3	Kaupalliset logiikat.....	17
<b>3</b>	<b>Dieselmoottorin pyörimisnopeussäätö .....</b>	<b>18</b>
3.1	Puunkatkaisukoneen ohjaaminen .....	18
3.2	Toteutus.....	19
3.3	Arviointikriteerit .....	21
3.4	Arviointi .....	22
<b>4</b>	<b>Lämminvesivaraajan latauspumpun ohjaus .....</b>	<b>25</b>
4.1	Saunan padan kiertovesipumpun ohjaus .....	25

4.2	Toteutus.....	26
4.3	Arviointi .....	28
5	<b>Autopilotti</b> .....	29
5.1	Veneen automaattiohjaus.....	29
5.2	Toteutus.....	31
5.3	Arviointi .....	34
6	<b>Tulokset</b> .....	36
7	<b>Pohdinta</b> .....	37
	Lähteet.....	39
	Liitteet .....	40
	Liite 1. Plazic käyttöohje.....	41

## Kuviot

Kuvio 1.	RC-servo ohjaa dieselmoottorin kaasuvipua.....	10
Kuvio 2.	Kaksi erilaista Arduino Mega-tyyppistä logiikkaa.....	11
Kuvio 3.	Logiikan ohjelmiston sisäinen rakenne.....	13
Kuvio 4.	Tietueen rakenne.....	15
Kuvio 5.	RS-toimilohko Plazicissa.....	15
Kuvio 6.	Sarjaliikennepääteohjelma.....	16
Kuvio 7.	Induktiivinen anturi asennettuna paikalleen .....	19
Kuvio 8.	Kahdet eri säätöparametrit .....	20
Kuvio 9.	Asetusarvon muuttaminen painonapeilla.....	20
Kuvio 10.	ESD-5550 -säädin.....	23
Kuvio 11.	Pata, josta lähtee putket varaajalle ja pumpulle.....	25
Kuvio 12.	Lämpötilavertailu.....	26
Kuvio 13.	Pumpun intervalliohjaus.....	26
Kuvio 14.	Latauspumpun ohjausyksikkö ulkoa ja sisältä.....	27
Kuvio 15.	Latauspumpun ohjaukseen käytettävä ohjausyksikkö Sorel STDC.....	28
Kuvio 16.	Pinnopilotti asennettuna.....	30
Kuvio 17.	Garmin Compact Reactor-järjestelmä.....	30

Kuvio 18. 3D tulostettu toimilaite.....	31
Kuvio 19. PWM-ohjain.....	31
Kuvio 20: Kompassimoduuli kytketty I2C väylään.....	32
Kuvio 21. Kompassi-toimilohko Plazic-sovellusohjelmassa.....	33
Kuvio 22. Simrad TP10-ohjain.....	34
Kuvio 23. Simrad-laitteet väylässä.....	35

## **Taulukot**

Taulukko 1. Erilaisia Arduino-piirikortteja ja niiden tärkeimmät ominaisuudet.....	6
Taulukko 2. Ulkoiset I/O-liitännät.....	21
Taulukko 3. Dieselmoottoriohjaimien vertailu.....	24
Taulukko 4. Latauspumppuohjaimien vertailu.....	29
Taulukko 5. Autopilot-ohjaimien vertailu.....	36

# 1 Plazic-ohjelma automaatiokäytössä

## 1.1 Plazic-ohjelmisto

Opinnäytetyön aiheena oli Plazic, Arduino-pohjaisen ohjelmoitavan logiikan vertailu kaupallisiin ohjausjärjestelmiin. Plazic ohjelmisto on Arduino-mikrokontrolleriin aiemmin tekemäni ohjelmisto. Ohjelmisto on tehty yleiskäyttöiseksi logiikkaohjelmaksi. Ohjelmistolla on tehty muutamia käytännön toteutuksia. Opinäytetyössä vertailtiin Plazic-ohjelmistolla tehtyjä toteutuksia kaupallisiin ohjausjärjestelmiin. Työllä ei ole ulkopuolista toimeksiantajaa.

## 1.2 Tavoite

Työn tarkoituksena oli tuoda automaatioalalle uutta näkökulmaa ohjaustekniikkaan. Nykyaikaiset automaation ohjausjärjestelmät ovat kehittyneet erittäin monipuolisiksi, eikä Plazic-sovellus pysty kilpailemaan tällä alueella. Arduino-mikrokontrollerit ovat kuitenkin edullisuudellaan ja monipuolisuudellaan innoittaneet harrastajia tekemään erityyppisiä käytännön sovelluksia. Valmiiden Arduinon sovellusohjelmien tarjonta on vähäistä. Ohjelmaesimerkkejä ja projektitoteutuksia on paljon. Plazic-sovelluksen ideana on tarjota mahdollisuus käyttää yksinkertaista ja edullista Arduino-ohjausta kevyempiin ohjattaviin kohteisiin. Tavoitteena oli selvittää Plazic-sovelluksen käyttökelpoisuutta käytännön ohjauskohteissa.

## 1.3 Tutkimusaineisto ja menetelmät

Vertailu toisiin kaupallisiin ohjausjärjestelmiin päätettiin toteuttaa case-tutkimus tyyppisesti. Ohjausjärjestelmien valintaan vaikuttaa useat eri tekijät ja case-tutkimustapa mahdollistaa eri näkökulmien käsittelemisen ohjausjärjestelmien vertailussa. Työssä päädyttiin vertailemaan kolmea erityyppistä toteutusta, joista kaksi ensimmäistä on jo toteutettu ja kolmannen toteutus on vielä kesken. Dieselmoottorin pyörimisnopeussäätö oli ensimmäinen PI-säätimen sisältänyt työ.

Toinen työ, lämminvesivaraajan latauspumpun ohjaus, sisältää paljon analogiasignaalien käsittelyä. Kolmas toteutus, veneen autopilotin ohjelmisto, sisältää yhteyden ulkoiseen kompassiin. Vertailukohteeksi valittiin tyypillinen nykyaikainen logiikka, Siemens S7 1200. Lisäksi tein vertailua edellä mainittuihin kohteisiin kaupallisesti myynnissä olevilla ohjausyksiköillä.

Opinnäytetyön lähtökohtainen ongelma oli, miten Plazic-sovellusta voi hyödyntää käytännön kohteissa. Näkökulmaa laajennettiin myös laitteistoon, koska ohjelmisto vaatii aina laitteiston toimiakseen. Sovelluksen hyödyllisyyttä arvioitaessa oli tarkoitus saada kokonaisarvio laitteen soveltuvuudesta, käyttäjäystävällisyydestä, hinnasta ja muunneltavuudesta. Ongelman käsittely rajattiin kolmeen tutkittavaan kohteeseen. Tutkimuksen oli tarkoitus tuottaa arvio sovellusohjelman käyttökelpoisuudesta. Työn tarkoituksena oli myös antaa ideoita, miten Plazic-ohjelmistoa voisi kehittää.

## 2 Arduino-pohjainen automaatio

### 2.1 Mikroprosessoripohjainen automaatio

Nykyaikainen automaatiolaite sisältää käytännössä aina mikroprosessorin. Releohjaus ja elektroniikalla ilman prosessoria toteutetut ohjausjärjestelmät ovat erittäin pienten ohjattavien kohteiden ohjausjärjestelmiä. Prosessorin suorittama ohjelma mahdollistaa erittäin monimutkaiset toiminnot ja ohjelmistoja muuttamalla toimintoja voi muokata lähes rajattomasti. Elektroniikkakomponenttien hinta on laskenut niin matalaksi, että prosessoritekniikalla toteutettuja ohjauksia löytyy lähes kaikkialta. (Keinänen, Kärkkäinen, Metso & Putkonen 2002, 206-213.)

Mikroprosessorien ja –kontrollerien ohjelmointi, kytkentä ja suunnittelu vaatii syvällistä osaamista. Massimo Banzi ja David Cuartielles aloittivat Arduino-nimisen projektin, jonka tarkoituksena oli kehittää opiskeluympäristö mikrokontrollerien opiskeluun. Projekti on tuottanut useita erilaisia Arduino-laitteita, Arduino IDE - ohjelmointityökalun ja huomattavan määrän lisälaitteita. Arduinosta onkin muodostunut käsite alan harrastajille. (Heinisuo 2012, 30-35.)



Mikroprosessoripohjainen laite jaetaan tavallisesti laitteistoon ja sen sisällä toimivaan ohjelmistoon. Fyysistä elektroniikkaa kutsutaan hardwareksi, kovoksi tai ammattislangilla raudaksi. Arduinossa laitteisto muodostuu erityyppisistä piirikorteista, joihin on asennettu mikrokontrolleri oheislaitteineen (ks. taulukko 1).

Taulukko 1. Erilaisia Arduino piirikortteja ja niiden tärkeimmät ominaisuudet (Compare board specs n.d.).

Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
<b>Mega 2560</b>	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
<b>Micro</b>	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
<b>MKR1000</b>	SAMD21 Cortex-M0+	3.3 V / 5V	48MHz	7/1	8/4	-	32	256	Micro	1
<b>Pro</b>	ATmega168	3.3 V / 3.35-12 V	8 MHz	6/0	14/6	0.512	1	16	-	1
	ATmega328P	5 V / 5-12 V	16 MHz			1	2	32		
<b>Pro Mini</b>	ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	1	2	32	-	1
<b>Uno</b>	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
<b>Zero</b>	ATSAMD21G18	3.3 V / 7-12 V	48 MHz	6/1	14/10	-	32	256	2 Micro	2
<b>Due</b>	ATSAM3X8E	3.3 V / 7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4

Plazic sovellus vaatii vähintään Arduino Mega 2560:n ominaisuudet toimiakseen. Arduino-piirikortit sisältävät yleensä USB-liitännän tietokoneelle ohjelmointia varten, syöttöjännitteen reguloinnin, mikrokontrollerin oheiskomponentteja, sekä liittimet I/O-liitäntöjä varten. Arduino-kortit poikkeavat toisistaan huomattavasti muistinkoon, I/O-liitäntöjen ja prosessorin suorituskyvyn osalta (ks. taulukko 1).

Mikroprosessorissa ja -kontrollerissa on aina ulkoisia liitäntöjä. Prosessori käyttää kommunikointiin väyläliitäntää, jonka avulla tieto siirtyy prosessorin, muistien ja I/O-yksiköiden välillä. Mikrokontrollerissa on yhdistetty prosessori, muistit ja I/O-yksiköt samaan fyysiseen komponenttiin. Arduino Mega 2560 käyttää ATmega2560-tyyppistä mikrokontrolleria. Tämä rakenne mahdollistaa yksinkertaisen ja edullisen tavan toteuttaa ohjausjärjestelmiä. Hardware voidaan suunnitella ohjaamaan vain yhtä tiettyä laitetta. Jos mikrokontrolleri on suunniteltu ja asennettu laitteeseen kiinteästi oheislaitteineen, kutsutaan sitä yleensä sulautetuksi järjestelmäksi.

Sulautetulle järjestelmälle on ominaista, että käyttäjä ei välttämättä ajattele käyttävänsä tietokonetta ja ohjelmistoympäristöt ovat suljettuja. Arduino-laitteistolla voidaan toteuttaa sulautettuja järjestelmiä, mutta avoin ohjelmointiympäristö mahdollistaa paljon muutakin. Mikrokontrollerin fyysisiin liittämiin kytketään yleensä käyttöjännite, kide, I/O-kytkennät ja mahdollisesti joitakin muuta riippuen kontrollerityypistä. (Ihalainen n.d.)

## 2.2 I/O-liitännät

Mikrokontrollerin I/O-liitännöillä tarkoitetaan Input- ja Output-liitäntöjä. Tyypilliseen ohjausjärjestelmään tuodaan kaikki ohjattavan prosessin informaatio prosessorille. Prosessori suorittaa oman algoritminsa avulla päätelmät ja ohjaa prosessia toimilaitteiden avulla. Mikrokontrollerille välitettävä tieto voi olla digitaalista, analogista tai väyläpohjaista. Mikrokontrolleri ohjaa prosessia analogisen, digitaalisen tai väyläpohjaisen informaation välityksellä. Arduino käyttää näitä kaikkia informaation siirtotapoja ja Plazic-sovellus mahdollistaa näiden helpon hyödynnettävyyden ohjelmistossa. (Kippo&Tikka 2008, 49)

### 2.2.1 Digitaalitulo

Digitaalisen tuloliitännän kautta voidaan välittää mikrokontrollerille päälle/pois-tieto. Liitäntää kutsutaan nimellä Digital Input tai Binary Input ja lyhennetään kirjaimilla DI. Kun mikrokontrollerin DI-liittimen jännite nousee tietyn rajan yläpuolelle, tulkitsee mikrokontrolleri DI-kanavan olevan tilassa "1". Jänniterajan alapuolella oleva jännite tulkitaan tilaksi "0". Ohjelmisto voi käyttää tilamuutoksen informaatiota. Jänniterajoina käytetään yleensä mikrokontrollereissa TTL-logiikan jännitetasoja. Käytännön kytkennöissä jännitetasot voivat olla esimerkiksi 24 V, jolloin jännitetasot pitää sovittaa sopiviksi. Sähköiset häiriöt ja maapotentiaalierot lisäävät vaatimuksia I/O-liitäntöjen toteuttamiselle. Paras häiriönsietoisuus saadaan toteuttamalla I/O-liitäntä optoerottimen avulla. Tällöin mikrokontrolleri saadaan täysin galvaanisesti erotettua ulkoisista piireistä. Optoerottimessa informaatio siirretään valon avulla. Toinen DI-tulon tärkeä ominaisuus on sen nopeus. Informaatio voi olla luonteeltaan hidasta, jolloin on tarpeen suodattaa nopeat häiriöt pois. Kytkin voi tuottaa yhden päällekytkemisen yhteydessä useita kytkentöjä kytkinvärähtelyn takia, joka sotkee

ohjelmiston toiminnan. Myös nopeita muutoksia pitää pystyä lukemaan. Pulssianturi tuottaa esimerkiksi 1024 kytKentäpulsia kierroksella. Moottori pyörittää anturia isolla nopeudella, joten taajuudet nousevat kHz-luokkaan. Plazic- ohjelmiston DI-tulolle on mahdollista parametroida lukuviive, jolla saadaan suodatus tulokanavalle. Isojen taajuuksien lukeminen tehdään keskeytystulon avulla. Normaalisti ohjelmisto lukee DI-kanavan tulon aina yhden ohjelmakierron aikana. Ohjelmakierto kestää kuitenkin liian kauan, jotta isoihin lukutaajuuksiin päästäisiin. Keskeytystulo keskeyttää prosessorin ohjelman, ja prosessori alkaa suorittamaan keskeytysohjelmaa eli ohjelmaa, jolla luetaan DI-tulon tila. Kun keskeytysohjelma on suoritettu, palautuu prosessori suorittamaan normaalia ohjelmaa. Plazic-sovellus käyttää keskeytystuloa Freq-toimilohkossa. (Kippo&Tikka 2008, 53.)

### 2.2.2 Digitaalilähtö

Digitaalisen lähtöliitännän kautta voidaan tehdä ohjauksia mikrokontrollerilla. Erilaisia kohteita voidaan ohjata päälle tai pois. Liitäntää kutsutaan nimellä Digital Output tai Binary Output, ja se lyhennetään yleensä kirjaimilla DO. Mikrokontrolleri pystyy ohjaamaan hyvin pientä virtaa. Arduinon maksimi ohjausvirta on 40 mA, ja siinä on tiettyjä rajoitteita. Jännite on 5 V. (Atmel 8-bit microcontroller 2011.) Isompia kuormia ohjattaessa käytetään tehotransistoria, MOSFET-ohjausta tai releä. Korkeampia jännitteitä ohjattaessa on syytä tehdä galvaaninen erotus mikrokontrollerin ja ohjattavan piirin välille. Galvaaninen erotus tehdään tavallisimmin optoerottimella tai releellä. (Kippo&Tikka 2008, 53.)

### 2.2.3 Analogiatulo

Analogiasignaalien avulla välitetään portaattomasti muuttuvaa informaatiota. Sähköinen viesti on verrannollinen siirrettävän informaation suuruuteen. Esimerkiksi mitattavan paineen suuruus voidaan välittää jänniteviestillä 0-10 V, niin että 0 V vastaa 0 Bar:n painetta ja 10 V vastaa 10 Bar:n painetta. Automaatiossa käytetään tavallisesti niin kutsuttuja standardiviestejä. Käyttämällä lähettävässä ja vastaanottavassa päässä samaa standardiviestiä, on laitteiden yhteensovittaminen helpompaa. Tyypillisiä virtaviestejä ovat 4...20 mA ja 0...20 mA. Virtaviestin hyviä puolia ovat häiriönsieto ja 4...20 mA viestissä mahdollisuus siirtää mittalähettimen

käyttöenergia samoilla mittajohtimilla. Jänniteviesteistä yleisimpiä ovat 0...10 V ja 0...5 V. Myös muita harvemmin käytettyjä standardiviestejä on olemassa. Muuntimilla voidaan muuttaa viestityyppi toiseksi. Pieni mitattu signaali muutetaan standardiviestiksi mittalähettimissä. Arduinon analogiaviesti on 0...5 V tai 0...3.3 V jänniteviesti riippuen Arduinon käyttämästä mikrokontrollerista. (Kippo&Tikka 2008, 50.)

Jos analogista informaatiota halutaan siirtää mikrokontrollerille, se tapahtuu analogiatulokanavan avulla. Kanavaa kutsutaan tavallisesti nimellä Analog Input, ja lyhennetään kirjainlyhenteeksi AI. Mikrokontrollerin AI-kanava on aina jännitemittaus. Tulokanavan kytkennässä voidaan muuttaa virtaviesti jänniteviestiksi resistanssin avulla. Mikrokontrollerin sisällä on A/D-muunnin, joka muuttaa jänniteviestin prosessorin ymmärtämäksi digitaalseksi viestiksi. A/D-muuntimelle ilmoitetaan muunnoksen resoluutio. Esimerkiksi Arduino Mega 2560-kortin A/D muunnin on 10-bittinen, eli 5 V:n jännitesignaali on jaettu 1024 osaan, jolloin pienin erottelukyky on 4.9 mV. Lämpötilamittaus NTC-vastuksen avulla tapahtuu kytkemällä sarjaan kiinteä vastus ja lämpötilariippuvainen vastus. Jännitejako muuttuu lämpötilan muuttuessa. Mittauksen tarkkuus jää huonoksi, koska jännitemuutos jää pieneksi, eikä koko 0...5 V A/D-muuntimen jännitealuetta hyödynnetä. (Kippo&Tikka 2008, 50)

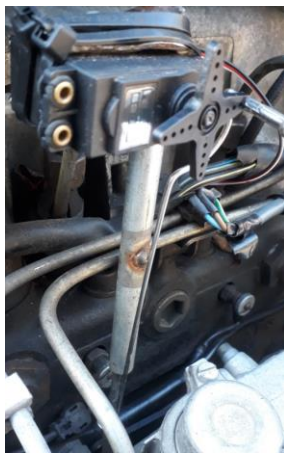
#### 2.2.4 Analogialähtö

Jos analogisella informaatiolla halutaan ohjata jotakin, käytetään analogista lähtökanavaa eli Analog Output-liitäntää. Lähtöä merkataan yleensä kirjaimilla AO. Prosessorin suorittama ohjelma antaa käskyn D/A-muuntimelle, joka muuttaa prosessorin digitaalisen käskyn analogiseksi jänniteviestiksi. D/A-muuntimen resoluutio kertoo, kuinka monessa portaassa muutos tapahtuu esimerkiksi 0...5 V:n välillä. Arduinon käyttämät mikrokontrollerit käyttävät PWM-tekniikkaa AO-kanavan ohjaamiseen. PWM-pulssinleveysmodulaation ideana on ohjata digitaalilähtökanavaa nopealla taajuudella päälle ja pois. Ohjaamalla pulssisuhdetta 0:n ja 100% välillä saadaan esimerkiksi RC-suotimella jänniteviesti aikaan. PWM-ohjausta voi käyttää suoraan esimerkiksi tehonsäädössä käyttämällä apuna FET-

pääteastetta. Veneen autopilotin toimilaitteen ohjaus onkin toteutettu tällä tekniikalla.

### 2.2.5 Muut liitännät

Arduinolla voi ohjata myös niin sanottuja RC-servoja (ks. kuvio 1). Radio-ohjatuissa pienoismalleissa käytetään toimilaitteina RC-servoja. Servo tarvitsee toimiakseen vain käyttöjännitteen ja asetusarvon. Asetusarvo annetaan PWM-ohjauksena. Noin 50 Hz taajuudella annetaan servolle pulsseja. Pulssin pituus kertoo servolle asetusarvon. Dieselmoottorin pyörimisnopeuden säätötyössä käytettiin kaasuvivun ohjaukseen RC-servoa.



Kuvio 1. RC-servo ohjaa dieselmoottorin kaasuvipua.

Mikroprosessoripohjaiset järjestelmät kommunikoivat perinteisten I/O-liitäntöjen lisäksi myös väylien avulla. Väylien avulla voidaan siirtää tietoa kahden tai useamman laitteen välillä. Arduino Mega 2560-kortilla on USB-portti, neljä sarjaporttia ja I2C- ja SPI-väylä. USB-portti varaa yhden sarjaportin käyttöönsä, mutta samaa porttia voi käyttää Bluetooth-moduulin kanssa. Arduinoon voidaan väylien avulla kytkeä esimerkiksi GSM-moduuli, reaaliaikakello, kompassimoduuli, GPS-yksikkö ja erilaisia sensoreita. Erilaisilla lisämoduleilla on mahdollista kytkeä Arduino Ethernetiin tai vaikkapa Modbussiin RS485-väylämuuntimen kautta. Plazic-sovellukseen on tehty ohjelmistorajapinta reaaliaikakellolle, kompassimoduulille ja GSM-modeemille. Bluetooth-moduuli muuntaa vain sarjaliikenteen langattomaan muotoon, eikä sille ole tehty erillistä ohjelmistoa.

### 2.2.6 Arduino-kokoonpanon toteutus

Arduino-pohjaisen ohjausjärjestelmän toteutus tehdään Arduino-piirikorteilla, liittämällä siihen tarvittavat I/O-liitäntäkortit, virtalähde, kotelo ja mahdolliset muut tarvikkeet. Toteutus vaatii paljon kytkemistä, ja laitteiston sähköisen toteutuksen luotettavuuskin voi olla kyseenalainen, ainakin haasteellisemmissa ympäristöolosuhteissa. Markkinoille on tullut muutamia teollisuuskäyttöön tarkoitettuja Arduino-pohjaisia logiikoita, joissa laitteisto on rakennettu valmiiksi. Logiikoihin on integroitu tavallisimmat I/O-sovittimet, väyläliitännät ja mahdollisesti näyttöyksikötkin (ks. kuvio 2).



Kuvio 2. Kaksi erilaista Arduino Mega-tyyppistä logiikkaa. (M-Duino PLC n.d.; Digital Loggers PLDuino n.d.)

### 2.3 Mikroprosessorin ohjelmisto

Mikroprosessori tarvitsee aina ohjelmiston toimiakseen. Mikroprosessori ymmärtää vain konekielisiä käskyjä. Konekieliohjelmointi on kuitenkin haastavaa ja hidasta, koska konekielisten käskyjen suorittamat toimet ovat yksinkertaisia bittitason toimintoja. On ollut tarve kehittää korkeamman tason ohjelmointikieliä, joilla tuotetaan mikroprosessorille konekielistä koodia. Tällä tavoin ohjelmoinnin tuottavuus on saatu paremmaksi, ohjelmointi on helpompaa ja visuaalisuutta on saatu lisättyä. Lisäksi on kehitetty erilaisia ohjelmistoja ja ohjelmointikieliä erityistarpeisiin. Automaatioon on kehitetty erilaisia ohjelmointikieliä helpottamaan koneen ohjauksen suunnittelua. Automaatiolaittevalmistajat ovat kehittäneet omia ohjelmointikieliä, joista tunnetuimmat ovat LAD- ja FBD-ohjelmointikielet. LAD- eli Ladder Logic-tikapuuohjelmointi muistuttaa sähköpiirikaavion ulkonäköä. FBD- eli

Function Block Diagram lohko-ohjelmointikieli sisältää erilaisia toimintoja. Toiminnot kuvataan ohjelmassa laatikoilla, joden välille tehdään tavallisesti graafisella ohjelmointityökalulla kytkennät. Standardi IEC 61131-1 ja -3 on tehty ohjelmoitavia logiikkoja varten. Standardi määrittelee yhteensä kuusi erilaista ohjelmointikieltä. Plazic-ohjelma on tehty FBD-ohjelmointityyppiseksi. Ohjelmassa ei ole suoraan graafista ohjelmointityökalua, mutta visualisointiin voidaan käyttää muita apuohjelmia.

## 2.4 Ohjelmoitava logiikka

Automaatio jaetaan tavallisesti kahteen eri päähaaraan: kappaletavara-automaatioon ja prosessiautomaatioon. Prosessiautomaatiolle on tyypillistä analogiasignaalien käsittely ja säätötekniikka. Kappaletavara-automaatiossa on enemmän digitaalisia signaaleita. Automaation alkuaikoina laitteet olivat suunniteltuja toteuttamaan vain yhden tyyppisiä ohjauksia.

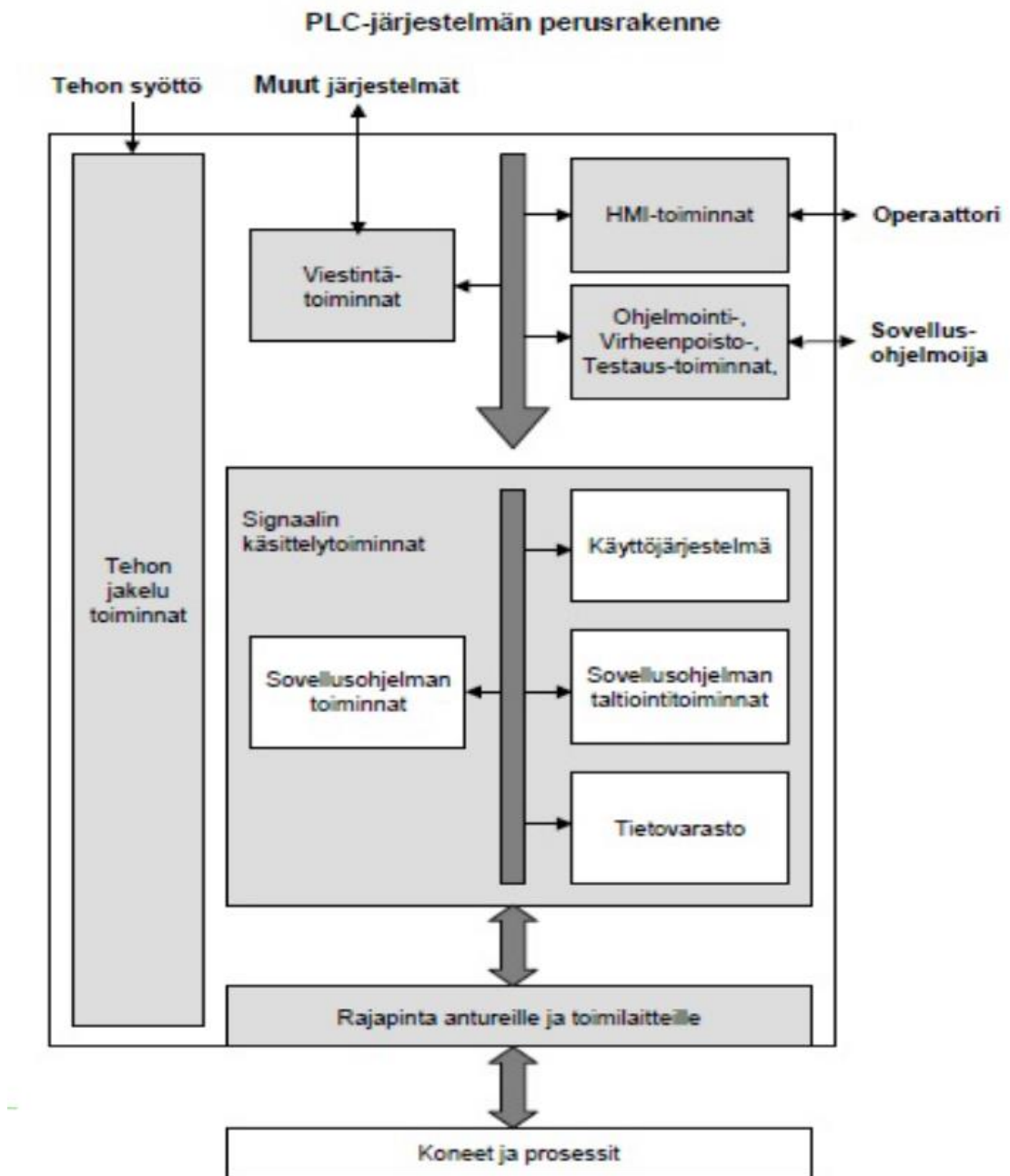
Automaatilaitevalmistajat ovat valmistaneet tyypillisesti itse omat tietokoneensa prosessiohjausta varten. Elektroniikka, tietokoneen rakenne ja liitännät ovat suunniteltu soveltumaan automaation käyttötarkoituksiin. Vasta viime vuosina on alettu käyttämään automaation ohjauksissa PC-tietokoneita. Valvomokäytössä PC-tietokoneita on ollut pidempään.

Prosessiautomaatiossa on ohjausjärjestelminä olleet prosessiautomaatiojärjestelmät. Prosessiautomaatiojärjestelmät ovat yleensä laajoja koko tehtaan kattavia järjestelmiä, jotka on rakennettu prosessiautomaation ohjauksia silmällä pitäen. Näissä järjestelmissä on mahdollista käyttää sekä analogisia että digitaalisia signaaleja.

### 2.4.1 Logiikan rakenne

Kappaletavara-automaation ohjauslaitteeksi on vakiintunut ohjelmoitava logiikka PLC, joka tulee sanoista Programmable Logic Controller. Pitkään logiikat olivat yksittäisiä itsenäisiä ohjaimia, joihin liitettiin vain digitaalisia liitäntöjä. Nykyaikaisen ohjelmoitavan logiikan ja prosessiautomaation raja on hukkumassa. Logiikoilla voidaan tehdä analogiamittauksia ja –ohjauksia. Laitteet voidaan kytkeä väylillä

yhteen ja logiikkaan on mahdollista liittää valvomotoiminnot. Fyysiset I/O-liitännät on mahdollista hajauttaa kentälle, lähelle ohjattavia kohteita hajautetun I/O:n avulla. Arduino-mikrokontrollerit sisältävät monipuoliset I/O-liitännät ja mahdollistavat vastaavien toimintojen toteuttamisen, mikäli ohjelmistosta löytyy tuki liitäntöjen käyttämiselle. Logiikan sisäisiä toimintoja voidaan kuvata kaaviolla.



Kuvio 3. Logiikan ohjelmiston sisäinen rakenne (Sundquist 2019).

PLC:n sisäinen ohjelmisto voidaan jaotella toimintojen mukaan (ks. kuvio 3). Käyttöjärjestelmä hoitaa perustoiminnot, liitäntäohjelmistot hoitavat ulkoista



liikennettä ja sovellusohjelma sisältää ohjattavaan kohteeseen räätälöidyn ohjelman.

Plazic-ohjelmiston perusohjelma sisältää:

- sarjaliikennepäätteellä operoitavan käyttöliittymän
- sovellusohjelman tulkin
- reaaliaikakellon liitännän
- GSM modeemiliitännän
- magneettikompassin liitännän
- I/O-liitäntöjen operoinnit

Sovellusohjelmaan on tehty avoin rajapinta kaikkiin Plazic-sovelluksen toimintoihin.

Sovellusohjelman avulla otetaan käyttöön tarvittavat toiminnot ja räätälöidään

ohjattavaa prosessia varten sopivat ohjaukset. Reaaliaikakellon ja

magneettikompassin ohjelmoinnissa on käytetty valmiita ohjelmakirjastoja apuna.

#### 2.4.2 Plazic-ohjelman toiminta

Plazic-sovellusohjelman ydin koostuu keskusmuistiin varatusta alueesta, johon

sovellusohjelma sijoitetaan. Sovellusohjelma sisältää erilaisia toimilohkoja, jotka

määrittävät toiminnot. Tyypillisiä toimilohkoja ovat mm. AND, OR, DI, DO, PID.

Ohjelmalohkojen kuvaukset löytyy liitteenä 1 olevasta käyttöohjeesta.

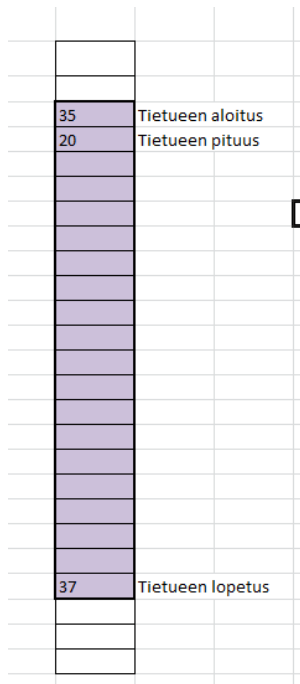
Perusohjelma aloittaa toimintansa etsimällä sovellusohjelman muistialueesta

toimilohkoja. Toimilohko on tallennettu muistialueelle tietueena (ks. kuvio 4).

Tietueen rakenne koostuu aloitusmerkistä, lopetusmerkistä ja tietueen pituudesta.

Jos muistialueesta löytyy tietue, suorittaa perusohjelma siihen toimilohkoon liittyvät

toiminnot. Muistin sisältö on 8-bit -luku, eli lukuarvo voi olla siis 0-255.



Kuvio 4. Tietueen rakenne

Jos esimerkiksi sovellusohjelmaan on tallennettu RS-toimilohko. Muistin sisältö näyttää kuvion 5 mukaiselta.

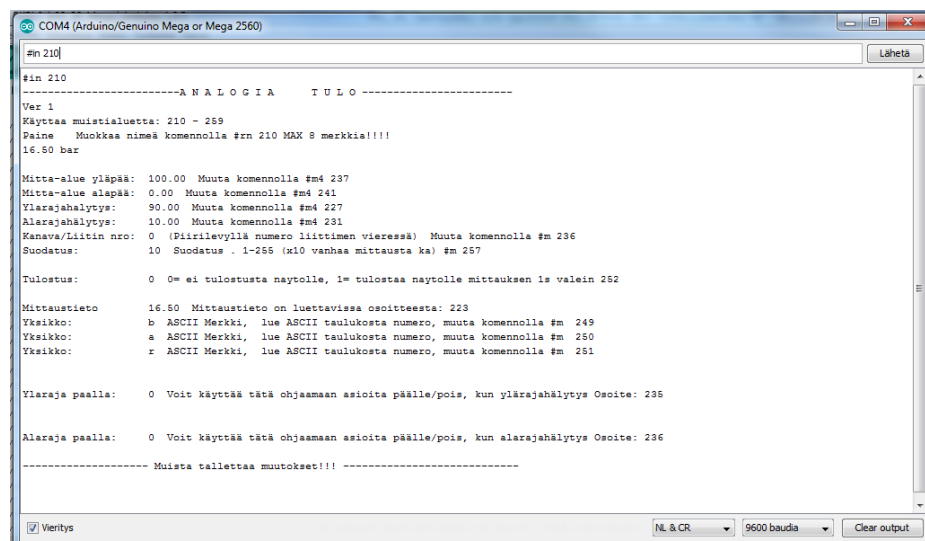
RS		
Muistitoiminto		
Rivi	Sisältö	
1	35	Tietueen aloitus
2	20	Tietueen pituus
3	82	RS Toiminto
4	1	Versio
5		Nimi: Merkki 1
6		Nimi: Merkki 2
7		Nimi: Merkki 3
8		Nimi: Merkki 4
9		Nimi: Merkki 5
10		Nimi: Merkki 6
11		Nimi: Merkki 7
12		Nimi: Merkki 8
13		End of text
14	100	SET Osoite (Low byte)
15		(High Byte)
16	101	RESET Osoite (Low byte)
17		(High Byte)
18		
19		Lähtö 0 / 1
20	37	Tietueen lopetus

Kuvio 5. RS-toimilohko Plazicissa

Muistialueelle on tallennettu vain sisältö-sarakkeen arvot. Perusohjelma löytää tietueen sijainnin aloitusmerkin, lopetusmerkin ja tietueen pituuden perusteella. Rivin kolme sisältö on 82 (ASCII merkki R) ja sen avulla perusohjelma alkaa suorittaa

RS-muistitoiminnon ohjelma-algoritmia. Ohjelma lukee rivien 14-15 osoitteesta SET-nastalle tilatiedon (muistipaikka 100). Riveiltä 16-17 löytyy osoite RESET-nastan tilatiedolle (muistipaikka 101). Näiden tilatietojen perusteella ohjelma ohjaa rivin 19 bitin päälle tai pois. Rivit 5-13 on varattu tietueen selväkieliselle nimelle. Rivin 4 versionumero on varaus tulevaisuuden ohjelmakehitystä varten. SET- ja RESET -nastojen tilat voidaan lukea esimerkiksi DI-toimilohkon tilatiedon osoitteesta. Jos RS-toimilohkolla halutaan ohjata DO-lähtöä, lisätään DO-toimilohko muistialueelle ja määritellään se lukemaan ohjaus RS-toimilohkon rivin 19 osoitteesta. Ohjelman toiminta on selvitetty perusteellisemmin käyttöohjeessa (ks. Liite 1).

Ohjelmistoon on rakennettu käyttöliittymä ohjelmointia ja operointia varten. Käyttöliittymäksi on valittu sarjaliikennepääteohjelma (ks. kuvio 6). Ohjelma lähettää Arduinolle näppäimistöltä lähetetyt merkit. Arduinon lähettämät merkit tulostetaan näytölle. Pääteohjelmana voidaankin käyttää melkein mitä tahansa pääteohjelmaa. Tämä mahdollistaa esimerkiksi matkapuhelimella Plazic-sovelluksen ohjaamisen, ohjelmoinnin ja parametrien muuttamisen.



Kuvio 6. Sarjaliikennepääteohjelma

Sovellusohjelmointiin ei ole graafista käyttöliittymää, mutta ohjelmoinnin helpottamiseksi on tehty Excel-taulukot toimilohkoista. Suunnittelemalla etukäteen ohjelma Excelin taulukoilla, saadaan ohjelmointi tehtyä helpommin, ja se tulee

samalla dokumentoitua. Valmis sovellusohjelma on mahdollista lähettää tiedostona Plaziciin, eikä jokaista muistipaikkaa tarvitse syöttää ohjelmaan käsin. Sarjaliikennesovelluksen kautta muokataan RAM-muistissa olevaa sovellusohjelmaa. Jos muutokset halutaan tallentaa pysyviksi, tallennetaan RAM-muistin sisältö EPROMiin, jolloin muutokset jää pysyviksi. Tämä mahdollistaa myös ohjelmamuutosten kokeilemisen, koska tallennusta ei ole pakko tehdä. Ihmisen ja logiikan välinen vuorovaikutus on mahdollista tehdä eri tavoin. Perinteinen HMI (Human Machine interface) tehtiin kytkimillä, merkkilampuilla, potentiometreillä tai analogianäyttöillä. Nykyinen näyttötekniikka tarjoaa graafisia käyttöliittymiä ja kosketusnäyttöjä. Graafisen näytön symbolit kertovat logiikkaohjelman tilatietoja, ja ohjaustoimenpiteet näytöltä lähtevät logiikkaohjelman läpi kenttälaitteille. Nykyaikaisilla logiikkavalmistajilla on kaikilla graafiset ohjauspaneelit käytössä. Laajemmat ja monimutkaisemmat HMI-liitännät toteutetaan tietokoneen valvomo-ohjelmistoilla. Arduinon on saatavissa useita erilaisia graafisia näyttöpaneeleita. Plazic-ohjelmasta puuttuu vielä tuki paneelikäytölle.

Plazic-ohjelmaan on tehty tavallisimmat digitaaliset toimilohkot: DI, DO, AND, OR, XOR, viiveet. Lisäksi ohjelmaan on tehty kello-ohjaus reaaliaikakellon perusteella ja GSM-hälytys ja ohjaustoiminto. Analogiatoimilohkojen toimintoja ovat AI, AO, raja-arvo, signaalin rajoitus, matemaattiset toiminnot, PI-säädin ja RC- servo-ohjaus. Omien toimilohkojen ohjelmointi on erittäin helppoa esimerkkien avulla. Toimilohkojen kuvaukset on kerrottu tarkemmin käyttöohjeessa. Plazic-ohjelman ulkoiset väylät ovat käytössä ainoastaan sarjaliikenneohjelmistolle, GSM-modeemille, reaaliaikakellolle ja kompassiyksikölle. Väyläliitännät ovat helppoja tehdä valmiiden ohjelmakirjastojen avulla. Automaatioväylistä ainakin Modbus-väylään on valmis ohjelmakirjasto olemassa Arduinolle. Myös valmiita liitäntäyksiköitä on olemassa, esimerkiksi Bluetooth-, Ethernet-, WLAN- ja RS485-liitäntöihin.

### 2.4.3 Kaupalliset logiikat

Opinnäytetyön määrittelyssä haluttiin tehdä vertailua johonkin tunnettuun kaupalliseen logiikkaan. Vertailukohteeksi valittiin Siemens S7 1200-sarjan logiikat.

Suomessa yleisesti käytettyjä logiikkavalmistajia ovat Omron, Mitsubishi, ABB, Fanuc, Beckhoff ja Siemens. Siemens S7-mallisarjan tuotteet tarjoavat erittäin monipuolisia ohjausjärjestelmäratkaisuja pienistä yksinäisistä logiikoista aina isoihin modulaarisiin järjestelmiin asti. Ethernet-pohjainen Profinet-väyläratkaisu mahdollistaa logiikoiden, HMI-laitteiden, taajuusmuuttajien ja muiden laitteiden yhteenliittämisen.

TIA-portal -ohjelmiston avulla voidaan järjestelmä konfiguroida tehokkaasti.

Logiikoilla voidaan toteuttaa turvalogiikkatoimintoja. Hajautetun I/O-ratkaisun avulla voidaan kaapelointia vähentää huomattavasti. Myös kenttälaitteita voidaan kytkeä suoraan esimerkiksi ASi-väylään. Logiikat tukevat erilaisia ohjelmointikieliä.

Voidaankin sanoa, että Siemens S7-logiikalla voi ohjata lähes mitä vain automaatiolaitteita. Rajoituksia voi tulla esimerkiksi eri valmistajien laitteiden yhteenliittämisessä. OPC-rajapinnan kautta sekin onnistunee ainakin ohjaustekniikan osalta.

### **3 Dieselmoottorin pyörimisnopeussäätö**

#### **3.1 Puunkatkaisukoneen ohjaaminen**

Plazic-sovelluksella on toteutettu dieselmoottorikäyttöisen puunkatkaisukoneen pyörimisnopeuden säätö. Giljotiini-tyyppisen puunkatkaisukoneen kuormitus muuttuu hetkessä suureksi, kun katkaistaan halkaisijaltaan isoa puuta. Perinteisen dieselmoottorin ruiskutuspumppuun on rakennettu mekaaninen keskipako-tyyppinen pyörimisnopeussäädin, jonka avulla esimerkiksi traktoria puun katkaisussa apuna käytettäessä tätä ongelmaa ei ole. Moottoriksi oli kuitenkin valittu henkilöauton dieselmoottori, jossa edellä mainittua ominaisuutta ei ole. Moottori sammui usein polttopuita tehdessä. Tehoa moottorissa olisi ollut, mutta sopiva säädin puuttui. Ongelmaa lähdettiin ratkaisemaan rakentamalla Arduino-pohjainen säädin, joka mittaa pyörimisnopeutta ja ohjaa kaasuvivun asentoa. Puun katkaisutapahtuma kestää alle sekunnin. Sääätöjärjestelmältä vaadittiin erittäin nopeaa reagointia. Pyörimisnopeuden mittauksen ja toimilaitteen aikavakioiden piti olla pieniä.

## 3.2 Toteutus

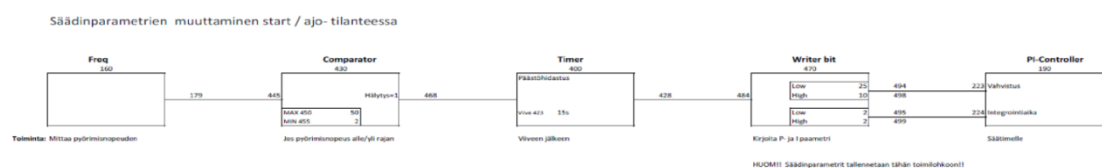
Pyörimisnopeussäädön toteutus tehtiin kesällä 2018. Tutkimuksessa käytettiin aineistona projektin suunnitteludokumentteja ja laitetoimittajien dokumentaatiota. Ohjausjärjestelmän toteutus on dokumentoitu erikseen. Pyörimisnopeuden mittaaminen tapahtuu pulssianturi-periaatteella. Kampiakselin päähän on asennettu induktiivinen lähestymiskytkin (ks. kuvio 7), joka antaa yhden pulssin jokaiselta kierrokselta. Pyörimistaajuuden mittaaminen voi tapahtua kahdella eri tavalla. Voidaan laskea tiettyä aikana tulleet pulssit tai mitata kahden pulssin välinen aika. Kokeilin kumpaakin tapaa, ja päädyin mittaamaan kahden pulssin välistä aikaa, ja laskemaan pyörimistaajuuden siitä. Tarkkailujakson olisi pitänyt olla pitkä, jotta mittaukselle olisi saatu edes jonkinlainen resoluutio. Induktiivisen anturin vaihtaminen pulssianturiksi olisi myös ollut mahdollista. Plazic-sovellukseen ohjelmoitiin Freq-toimilohko, joka käyttää Arduinon keskeytystuloa. Freq-toimilohkon muistipaikasta on luettavissa jokaisella ohjelmakierrolla uusi mittaustieto, jota voidaan tarvittaessa suodattaa. Ohjelmakierto Plazic-ohjelmassa kestää noin 0,02 s.



Kuvio 7. Induktiivinen anturi asennettuna paikalleen.

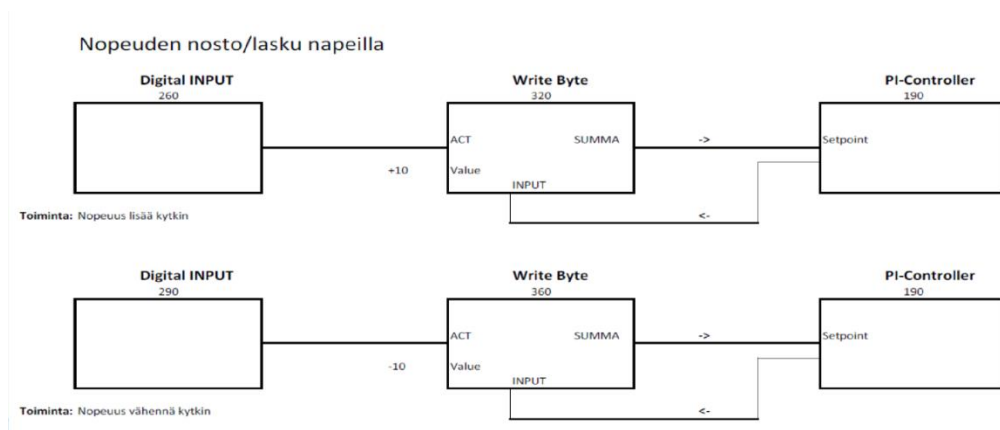
Toimilaitteeksi valitsin RC-servon. Radio-ohjattujen pienoismallien RC-servot ovat nopeita ja edullisia toimilaitteita. Esimerkiksi Futaba HPS-CB700 servon nopeus on valmistajan ilmoittamana 0.07s/60 ° ja vääntömomentti 49.0 kgf/cm 7.4 V syöttöjännitteellä. (HPS-CB700 Standard surface servo n.d.) Plazic:iin ohjelmoitiin PI-säädin.

Mekaanisen ruiskupumpun keskipakosäädin vastaa P-säädintä, joten arvelin, ettei D-algoritmia tarvittaisi. Perusohjelma suorittaa jokaisella ohjelmakierrolla säädintoimilohkon algoritmin. P-vaste lasketaan jokaisella ohjelmakierrolla. I-vaste vain 0,1s välein. Vasteet summataan yhteen, jolloin niistä muodostuu säätimen lähtösignaali. Säädintä virittäessä ongelmaksi muodostui kaksi erityyppistä tilaa, missä moottoria käytetään. Puunkatkaisussa säätimen vahvistuskertoimen pitää olla suuri. Käynnistystilanteessa, kun moottorin voimansiirron kytkin on vielä auki, oli säätimellä taipumus alkaa värähtelemään. Tämä ongelma ratkaistiin käyttämällä kaksia eri parametreja säätimellä (ks. kuvio 8).



Kuvio 8. Kahdet eri säätöparametrit

Koneen käyttäjälle haluttiin tehdä mahdollisuus muuttaa pyörimisnopeuden asetusarvoa. Asetusarvoa voi lisätä tai vähentää kahdella painikkeella (ks. kuvio 9).



Kuvio 9. Asetusarvon muuttaminen painonapeilla

Muut parametrien muutokset tehdään sarjaliikenneohjelmiston avulla. Esimerkiksi säätimen virittäminen tehtiin Bluetooth-yhteyden kautta matkapuhelimella puiden teon yhteydessä.

Taulukko 2 Ulkoiset I/O liitännät

Tarkoitus	Tyyppi
Lisää nopeutta-painike	Digital input
Vähennä nopeutta-painike	Digital input
Pyörimisnopeuden mitta	Pulssi tulo
RC-servon ohjaus	RC servon PWM ohjaus

### 3.3 Arviointikriteerit

Jos Plazic-sovellusta vertaillaan case-tutkimus tyyppisesti muihin tuotteisiin täytyy ensin määritellä käsitteet ja mittarit Kananen (2013, 130). Arviointikriteereiksi valitsin teknisen soveltuvuuden, käyttäjäystävällisyyden, hinnan ja muunneltavuuden.

Teknisellä soveltuvuudella tarkoitetaan sitä, soveltuuko laite teknisten ominaisuuksiensa puolesta ohjaamaan prosessia. Tärkeimpiä asioita ovat sähköinen yhteensopivuus, joka sisältää käyttäjänitteen ja I/O-kytkentöjen yhteensopivuuden ja ohjausominaisuuksien riittävyys, jolla tarkoitetaan ohjelmallisten ominaisuuksien monipuolisuutta sekä riittävän nopeaa reagointia säätö- ja ohjaustoiminnoissa. Arvioin teknisen soveltuvuuden asteikolla erinomainen, hyvä, tyydyttävä ja huono. Erinomainen tarkoittaa lähes täydellisiä ominaisuuksia prosessin ohjaukseen. Arvosana on hyvä, jos halutut ominaisuudet pystytään toteuttamaan, mutta se vaatii esimerkiksi lisätoimilohkojen ohjelmointia. Tyydyttävään arvosanaan jää pienehköjä puutteita. Huonolla ei pysty toteutusta tekemään.

Käyttäjäystävällisyydellä arvioidaan laitteen ohjelmoinnin helppoutta sekä laitteen loppukäyttäjän käyttäjäystävällisyyttä. Arvioin käyttäjäystävällisyyden asteikolla erinomainen, hyvä, tyydyttävä ja huono. Erinomainen tarkoittaa helppoa ohjelmointia ja käyttöönottoa. Myös loppukäyttäjän käyttöliittymä on selkeä ja helppokäyttöinen. Hyvän arvosanan saa, jos ohjelmoinnissa tai loppukäyttäjän käyttöliittymässä on puutteita. Tyydyttävän arvosanan rajana on, jos sekä



ohjelmoijan, että loppukäyttäjän toiminnot ovat vaikeasti toteutettavia. Huonon arvosanan saa, jos loppukäyttäjä ei pysty ilman erkoiskoulutusta toimimaan laiteympäristössä.

Hinta on kokonaisarvio kustannuksista. Arvioin hinnan asteikolla erinomainen, hyvä, tyydyttävä ja huono. Hinta-arvio on suhteellinen arvio, jossa eri toteutusten hinnat laitetaan järjestykseen suuruusjärjestyksessä. Halvin hinta saa arvosanan erinomainen, toiseksi halvin saa arvosanan hyvä ja kolmanneksi halvin tyydyttävä. Arviota huono käytetään, jos hinta nousee ohjattavan prosessin hintaan nähden merkittävän korkeaksi. Lopullisen hinnan määräytyminen riippuu useista eri tekijöistä, ja arviointi saattaa muodostua virheelliseksi. Toteutustavat saavat saman arvosanan, jos hinnat ovat 20 % sisällä toisistaan.

Muunneltavuusnäkökulma sisältää arvion laitteen laajennettavuudesta ohjaamaan jotakin muita lisätoimintoja. Arvioin ohjausjärjestelmät asteikolla erinomainen, hyvä, tyydyttävä, huono. Erinomainen muunneltavuus tarkoittaa monipuolisia ominaisuuksia ohjausjärjestelmässä, jotka on helppoja ottaa käyttöön. Hyvällä järjestelmällä voidaan toteuttaa lähes kaikki muutokset, mutta joitakin rajoitteita on tiedossa. Tyydyttävän järjestelmän laajennettavuudessa on paljon rajoitteita. Huonoa järjestelmää ei voida muuttaa.

### 3.4 Arviointi

Dieselmoottorin pyörimisnopeussäädön ohjaamista Plazic-ohjelmalla verrattiin kahteen erityyppiseen ohjausratkaisuun: Siemens S7 1200- logiikalla toteutettuun ohjaukseen ja ESD-5550 -säätimeen.



Kuvio 10. ESD-5550-säädin (ESD-5550/5570 Speed control unit manual n.d.).

Säädin on analoginen pyörimisnopeussäädin. Nopeustieto saadaan magneettiselta kierroslukuanturilta, joka asennetaan vauhtipyörän hammaskehän läheisyyteen. Toimilaitteena käytetään nopeaa säätimelle suunniteltua magneettista toimilaitetta. Asetusarvo annetaan laitteelle ulkoisella potentiometrillä. (ESD-5550/5570 Speed control unit manual n.d.)

Siemens S7 1200 toteutus tehdään samoilla ulkoisilla kytkennöillä, kuin vastaava Arduino toteutus. Sovellusohjelmaan otetaan käyttöön PID-säädin. Toimilaitteena voidaan käyttää RC-servoa ja erilaiset pulssianturit on mahdollista kytkeä logiikkaan. Ajoneuvon 12V DC-järjestelmän jännitetaso pitää nostaa 24V:iin sopivalla hakkurivirtalähteellä. Samalla virtalähde parantaa sähkön laatua. Virtalähteenä voisi käyttää esimerkiksi Eaton 12010-tyyppistä virtalähdettä.

Tekninen soveltuvuus eri laitteiden välillä on kaikilla erinomainen tai hyvä (ks. taulukko 3). ESD-5550 on suunniteltu tähän ohjaustarkoitukseen. Siemens logiikka monipuolisuutensa takia on myös erinomainen vaihtoehto. Plazic-sovelluksella on toteutettu dieselmoottoriohjaus onnistuneesti. Säätimestä puuttuu kuitenkin D-algoritmi, ainakin toistaiseksi. Nopeaa reagointia vaativissa säätötoiminoissa ominaisuus olisi tarpeen.

Käyttäjäystävällisyys on loppukäyttäjän kannalta kaikilla kolmella laitteella erinomainen. Käyttöönotto on kuitenkin ESD-5550 laitteella helpoin, koska ohjelmointia ei tarvita. Ohjelmoiminen tarvitsee aina erikoisosaamista.

Käyttöönottotilanteessa tarvitsee virittää säädinparametrit sopiviksi. Säädön hyvyys arvioidaan säätimen kyvystä pitää kierrosnopeus asetusarvossaan kuormitustilanteesta riippumatta. Kustannusten vertailu eri laitteiden välillä ei ollut yksiselitteistä. Siemens-logiikan hinta 1.5.2019 Elfalla on 367 e (alv 0%).

Plazic-sovelluksen käyttämä hardware voisi olla esimerkiksi PLC Arduino ARDBOX 20, jonka hinta ulkomaisessa verkkokaupassa on 141.75 e. Hinta Suomeen toimitettuna on kuitenkin lähempänä 200 e tulli- ja postikuluineen. Ohjelmointityökalut ja ohjelmointi maksavat jonkin verran. ESD5550 hinta on 122,36 e ulkomaisesta verkkokaupasta tilattuna, lisäksi täytyy tässäkin huomioida, että hinta nostavat tulli- ja postikulut. Toimilaitteen hinta on noin 100 e Suomeen toimitettuna. Myös magneettisen pyörimisnopeusanturin asentaminen vauhtipyörän hammaskehälle voi olla haastavaa. Ohjelmointikustannuksia ei kuitenkaan ole, joten kokonaiskustannuksiltaan ESD-5550- laite on edullisin.

Muunneltavuus ja laajennettavuus ovat tärkeitä kriteerejä ohjausjärjestelmiä suunniteltaessa. Dieselmoottorin ohjauksen mahdollisia tehtäviä olisi automaattinen käynnistys, automaattinen pysäytys öljynpaineen, yllilämmön tai polttoaineen vähyyden vuoksi. Ohjelmoitavilla laitteilla voidaan muutokset ja laajennukset toteuttaa, mutta ESD-5550- yksiköllä se ei ole mahdollista. Muunneltavuus Siemensin logiikassa on hyvien teknisten ominaisuuksien vuoksi paras. Plazic-sovelluksesta puuttuu toistaiseksi tuki graafiseen näyttöön, joka olikin ehkä suurin puute arvioitaessa muunneltavuutta. Näytön voisi korvata mobiililaitteella, mutta mobiililaitteeseen pitäisi tehdä käyttäjäystävällinen mobiilisovellus. Sarjaliikennesovelluksella voi tehdä kaikki toiminnot, mutta loppukäyttäjälle se ei ole sopiva työkalu.

Taulukko 3. Dieselmoottoriohjaimien vertailu

	Tekninen soveltuvuus	Käyttäjäystävällisyys	Hinta	Muunneltavuus
<b>Siemens S7 1200</b>	Erinomainen	Hyvä	Tyydyttävä	Erinomainen
<b>Plazic</b>	Hyvä	Hyvä	Erinomainen	Hyvä
<b>ESD-5550</b>	Erinomainen	Erinomainen	Erinomainen	Huono

## 4 Lämminvesivaraajan latauspumpun ohjaus

### 4.1 Saunan padan kiertovesipumpun ohjaus

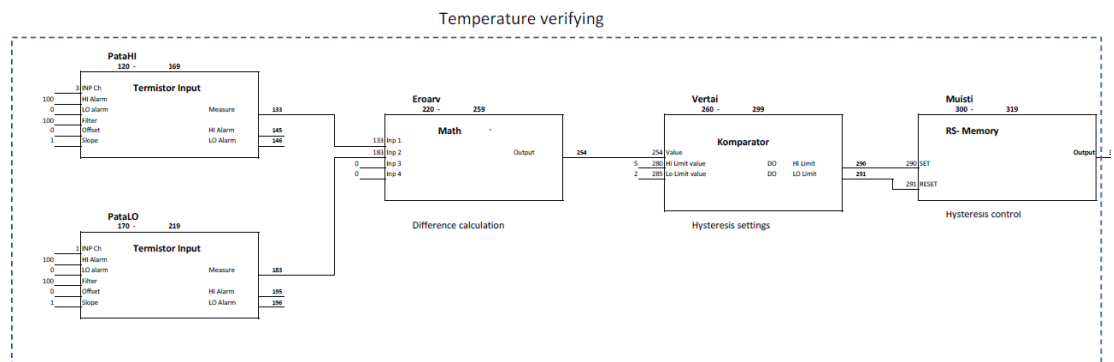
Rakennusten vesikiertoisessa lattialämmityksessä voidaan käyttää erityyppisiä lämmönlähteitä. Lämmönlähteitä voi kytkeä rinnakkain useampia ja lämmönlähde voidaan valita esimerkiksi energian hinnan tai ekologisuuden perusteella. Tässä opinnäytetyössä tutkimme Plazic-sovelluksella ohjattua järjestelmää, joka käyttää lämmitysenergianaan sähköä tai saunan padan lämmitysenergiaa (ks. kuvio 11).



Kuvio 11. Pata, josta lähtee putket varaajalle ja pumpulle

Saunan padassa poltetaan puuta ja tuli lämmittää avonaisessa astiassa olevaa vettä. Kuuma vesi johdetaan talon lämminvesivaraajassa olevaan lämmönvaihdinkierukkaan kiertovesipumpun avulla. Pumppua ei kuitenkaan kannata käyttää silloin, kun tulipesässä ei ole tulta. Energiaa halutaan siirtää vain padasta varaajaan, eikä päinvastoin. Päättely energian siirtymissuunnasta tehdään kahden eri lämpöanturin perusteella. Lämmönvaihdinkierukkaan menevän ja sieltä palaavan

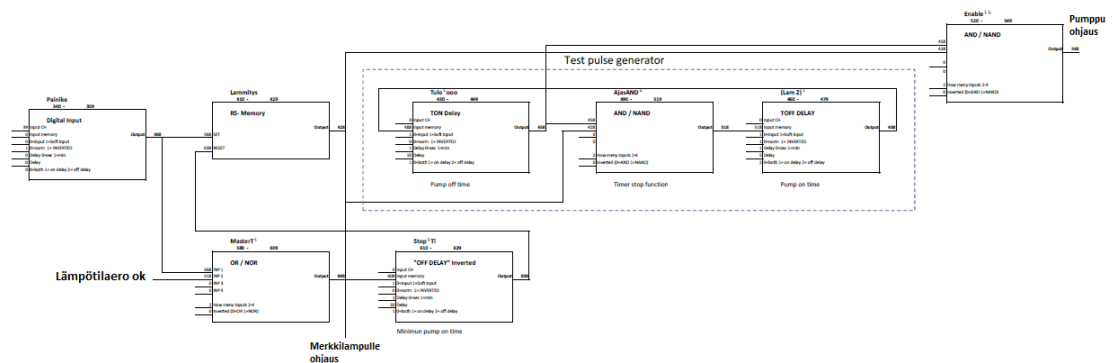
veden lämpötilat mitataan. Jos menevän veden lämpötila on korkeampi, on energian siirtosuunta oikea (ks. kuvio 12).



Kuvio 12. Lämpötilavertailu

Lämpöanturien mittaustulokset ovat luotettavia vain, jos vesi virtaa putkistossa.

Lämmityksen alkaessa painetaan käynnistys-painiketta. Pumppu käy intervallittyyppisesti, jotta ohjelma voi päätellä onko pata lämmennyt riittävästi. Kun riittävä lämpötila on saavutettu, pumppu jää päälle. Pumppu sammuu automaattisesti lämpötilan laskiessa tietyn rajan alapuolelle (ks. kuvio 13).

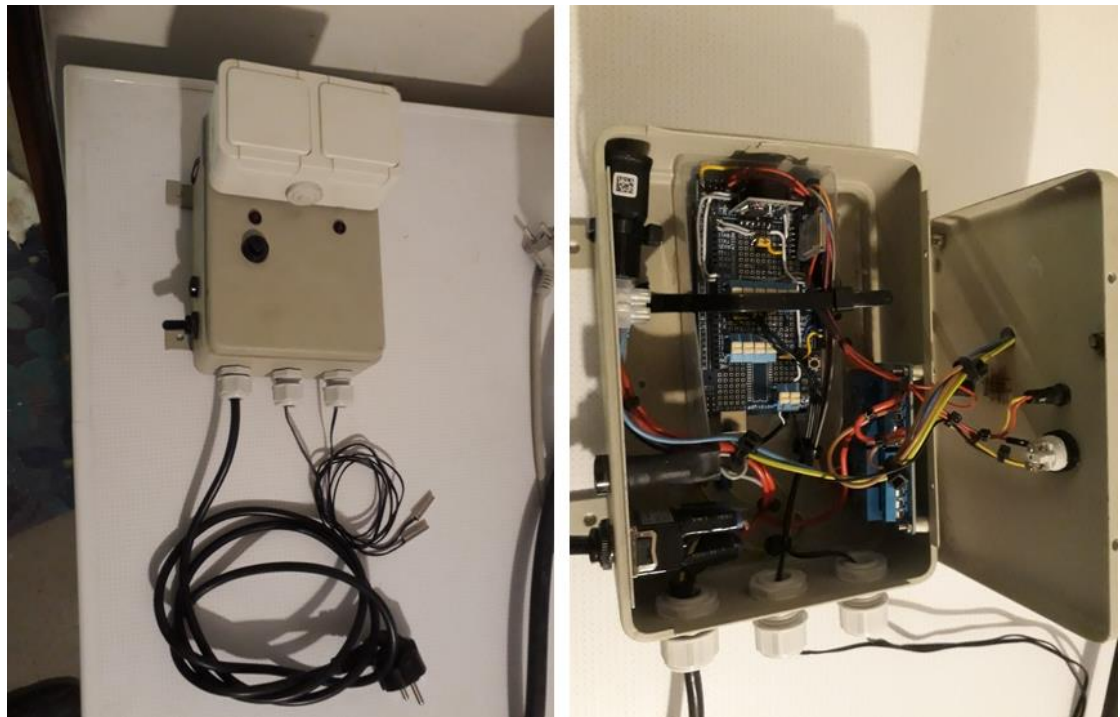


Kuvio 13. Pumpun intervalliohjaus

## 4.2 Toteutus

Sovellusohjelma tehtiin JAMK Automation technology laboratory work-kurssin miniprojekti-ryhmätyönä. Tutkimusaineistona käytettiin projektin suunnitteludokumentteja ja laitetoimittajien dokumentaatiota. Plazic-sovelluksella

tehty latauspumpun ohjaus on ollut lähes päivittäisessä käytössä kevättalven 2019. Lämpötilarajat ja viiveet on aseteltu matkapuhelimen sarjaliikennesovelluksella ja antureiden lämpötilat on luettavissa matkapuhelimella. Ohjaustapa on todettu hyväksi ja helppokäyttöiseksi. Kytkenät tehtiin metallikoteloon, johon sijoitettiin virtalähde, releyksikkö, Arduino Mega 2560, Bluetooth kortti ja RTA reaaliaikakello. (ks. kuvio 14). Reaaliaikakello asennettiin varalle, jos toisen pistorasian ohjaukseen tarvitaan aikaohjausta. Jos halutaan käyttää sähkötehon rajoitusta ajastettuna, voisi tehonrajoituksen ohjauksen toteuttaa pistorasian kautta. Lämpötila-antureina käytetään NTC-vastuksia.



Kuvio 14. Latauspumpun ohjausyksikkö ulkoa ja sisältä

Siemens S7 1200 toteutus ulkoisten kytkentöjen ja ohjelman osalta on hyvin samantyyppinen Plazic-toteutuksen kanssa. Lämpötila-anturia varten pitää logiikkaan lisätä SM 1231 RTD signal moduuli. Anturityypiksi käy esimerkiksi PT1000 anturi. Lämpötilatiedot logiikasta on mahdollista saada HMI-näytön avulla. Raja-arvot ja viiveet olisi myös mahdollista muuttaa samasta näytöstä. Logiikka voisi ohjata 24V apureleen avulla kiertovesipumppua.

Latauspumpun ohjaukseen on kehitetty myös siihen käyttöön tarkoitettuja ohjausyksiköitä. Sorel-niminen yritys on valmistanut kiertovesipumppujen ohjausyksikön, jolla voi ohjata lämpötilamittausten perusteella pumppua (ks. kuvio 15). Ohjausyksikössä on erilaisia ohjausmoodeja erilaisia prosesseja varten. Ohjaimeen kytketään käyttöjännite, anturit ja pumppu. Laitteen asentaminen on yksinkertaista ja parametreja on vähän. Laitteen näytöstä on luettavissa lämpötilat ja pumpun ohjauksen tila. (Temperature difference controller n.d).



Kuvio 15. Latauspumpun ohjaukseen käytettävä ohjausyksikkö (Temperature difference controller n.d).

### 4.3 Arviointi

Laitteiden ominaisuuksia vertaillaan kappaleessa 3.3 määriteltyjen arviointikriteerien mukaan. Laitteiden teknistä soveltuvuutta vertailtaessa on Siemens logiikka ominaisuuksiltaan erinomainen. Kaikki ohjaukseen ja käyttöliittymään liittyvät toiminnot voidaan toteuttaa. Myös Sorel STDC ominaisuudet ovat erinomaiset, koska laite on suunniteltu tähän käyttöön. Plazic:n puute on graafinen käyttöliittymä. Ohjausominaisuudet ovat hyvät kaikissa laitteissa (ks. taulukko 4).

Käyttäjäystävällisyys on loppukäyttäjän näkökulmasta erinomainen kaikilla laitteilla. Käyttöönotto on helpointa Sorel STDC laitteella. Plazic ja Siemens tarvitsevat sovellusohjelman laitteeseen. Työmäärä kummankin laitteen ohjelmoinnissa on samaa luokkaa.

Hintaa vertailtaessa Arduino-pohjaisen Plazicin ja Sorel STDC laitteen hinta on likipitään sama. Sorel säätimen hinta antureineen on 163,2e (Alv 24%) . Siemens S7 1200 logiikan vertailuhinta on Elfalla 367e (Alv 0%). Analogiatulomoduulin hinta on 298e (Alv 0%). PT1000 anturin hinta on 14,01e / kpl. Siemensin logiikalla kokonaishinta nousee korkeammaksi, kuin muilla toteustavoilla.

Muunneltavuutta vertailtaessa Sorel STDC ohjain saa heikoimman arvosanan, koska lisäominaisuuksia siihen ei voi lisätä. Omakotitalon omistajaa kiinnostaa saavutettu kustannussäästö. Lämmitysenergian määrää halutaan mitata ja laskea puusta saatava lämmitysenergia. Järjestelmä mittaa varaajaan menevän ja palaavan veden lämpötilat. Kiertovesipumpun vesimäärää voidaan pitää vakiona. Energiamäärä on laskettavissa helposti näistä suureista. Myös muita ohjattavia kohteita lämmitysjärjestelmästä löytyy. Jos varaajaa lämmitetään sähköllä ja sähkön hinta muuttuu tunneittain, voidaan sähkönkulutusta leikata kalliin hinnan aikana. Siemens-logiikalla ja Plazic- sovelluksella räätälöinti voidaan tehdä helposti. Siemens-logiikan ominaisuudet ovat kuitenkin paremmat, ja siksi se saakin Plazic-sovellusta paremman arvion. Plazic-sovellukseen voidaan toteuttaa avoimen lähdekoodin avulla vapaasti lisäominaisuuksia.

Taulukko 4. Latauspumppuohjaimien vertailu

	Tekninen soveltuvuus	Käyttäjäystävällisyys	Hinta	Muunneltavuus
Siemens S7 1200	Erinomainen	Hyvä	Huono	Erinomainen
Plazic	Hyvä	Hyvä	Erinomainen	Hyvä
Sorel STDC	Erinomainen	Erinomainen	Erinomainen	Huono

## 5 Autopilotti

### 5.1 Veneen automaattiohjaus

Venettä on usein tarve ohjata pitkiäkin matkoja suoraan eteenpäin. Aallokon, tuulen ja veneen oma epästabilisuuden vaikutus vaatii ohjaustoimintoja. Tähän käyttöön on olemassa erityyppisiä automaattisia ohjaimia, joita kutsutaan autopilotiksi. Autopilotin toiminta perustuu mitatun kulkusuunnan ja halutun kulkusuunnan



muodostaman eroarvon perusteella säätöalgoritmin mukaan laskettuun ohjaukseen. Kulkusuunnan mittaukseen voidaan käyttää maan magneettikenttää mittaavia antureita, erilaisia gyroskooppeja, GPS-satelliittipaikanusta tai jopa tuulta tuuliperäsinratkaisussa. Kaikki mittalaitteet ovat omalla tavallaan hyviä. Autopilotille riittää suhteellinen mittaustieto kulkusuunnasta, koska ohjausalgoritmi käsittelee vain eroarvoa. Aluksen navigointiin tarvitaan tarkka kulkusuunta esimerkiksi magneettikompassilta tai GPS-satelliittinavigointilaitteelta. Toimilaitteena käytetään pinnaohjauksessa lineaarimoottoriratkaisuja ja ruoripilotissa sähkömoottoria tai hydraulikkaa.



Kuvio 16. Pinnapilotti asennettuna (Pelagic Autopilot 2017).

Käyttöliittymänä on yksinkertaimmillaan vain painokytkimet. Vene ohjataan käsin haluttuun kulkusuuntaan, ja kytketään autopilotti päälle. Autopilotti tallentaa kulkusuunnan asetusarvoksi, ja pitää kulkusuunnan asetusarvossaan ohjaamalla peräsimen asentoa. Yleensä autopilotissa on painokytkimet esimerkiksi 1 ja 10 asteen kurssimuutoksille. Kalliimmissa laitteissa on värinäytölliset käyttöliittymät, jossa näkyy asetus- ja mittausarvo sekä peräsinkulman asento (ks kuvio 17).



Kuvio 17. Garmin Compact Reactor järjestelmä (Marine autopilotit n.d.).

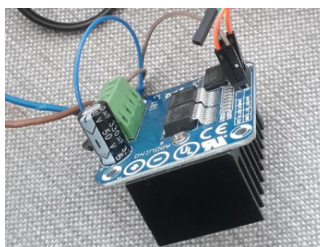
## 5.2 Toteutus

Oman autopilotin kehittäminen on lähtenyt liikkeelle purjeveneharrastuksesta. Yksin purjehdittaessa täytyy välillä säätää purjeita ja tehdä paljon muuta samalla, kun pitäisi ohjata venettä. Peräsimen lukitseminen paikalleen ei toimi riittävän hyvin. Toimilaitteeksi olen tehnyt lineaarimoottorin vanhasta akkuporakoneen moottorista ja kierretangosta (ks. kuvio 18). Moottorin ympärille on tehty 3D mallintamalla ja tulostamalla kuoret. Kierretangon suojaksi on asennettu sisäkkäin menevät teräsputket. Moottoriin on asennettu magneetti ja Reed-kytkin, josta saadaan moottorin asennosta suhteellista takaisinkytkentätietoa.



Kuvio 18. 3D tulostettu toimilaite

Moottoria ohjataan Arduinon PWM-ohjauksella. Moottori ottama virta on kuitenkin niin iso, että se vaatii moottoriohjaimen. Valitsin moottoriohjaimeksi Handson Technologyn valmistaman BTS7960 H-Bridge 43A ohjaimen (ks. kuvio 19). Ohjain voidaan kytkeä suoraan Arduinon DO-lähtöihin ja sillä voi tehdä myös moottorin pyörimissuunnan muutos.



Kuvio 19. PWM Ohjain

Anturivaihtoehtoina oli maan magneettikenttään perustuva anturi tai GPS-vastaanotin. Päädyin Honeywellin valmistamaan magnetoresistiiviseen X-Y-Z

HMC5883L-TR anturiin (ks. kuvio 20), koska anturi oli helposti liitettävissä I2C-väylän kautta Arduinoon ja siihen löytyi valmis ohjelmakirjasto. Ohjelmakirjastolla tarkoitetaan kommunikointiohjelmaa Arduino IDE:n ja anturin välillä, joka välittää suoraan kompassisuunnan Arduinon omaan ohjelmistorajapintaan. Plazic-sovellukseen on tehty oma toimilohko, joka välittää kompassisuunnan vielä Plazic:n analogiasignaaliaksi, jolloin sitä voi käyttää vaikka säätimen mittaustietona.



Kuvio 20: Kompassimoduuli kytketty I2C väylään

Kompassisuunta määritellään asteina pohjoisesta myötäpäivään. Ohjelman on käsiteltävä tilanne, jossa esimerkiksi asetusarvo on 359 astetta ja mittausravona on 1 aste. Eroarvoa ei voi laskea lukuarvojen erotuksena, koska tällöin laskutoimitukseen tulee täyden kierroksen virhe. Ratkaisin ongelman lisäämällä kompassi-toimilohkoon eroarvon laskennan. Eroarvoa voi käyttää suoraan PI-säätimen asetusarvona. Toimilohkoon tehdään vielä asetusarvon +/- toiminnot tulevaisuudessa, koska siinäkin pitää ottaa huomioon täyden kierroksen aiheuttama ongelma. Kompassilukemaan ei ole ollut tarve tehdä eksymän ja erannon aiheuttamia korjauksia, koska lukemaa käytetään vain suhteellisen poikkeaman määrittämiseen tallennetusta kurssista. Toimilohkosta on luettavissa kompassisuunta ja eroarvo, jos asetusarvo on annettuna (ks. kuvio 21).

Kompassi							
Lukee kompassilta suunnan.							
1	#	35	Tietueen aloitus				
2	40	40	Tietueen pituus				
3	75	K					
4	1	1	Versio				
5			Nimi: Merkki 1				
6			Nimi: Merkki 2				
7			Nimi: Merkki 3				
8			Nimi: Merkki 4				
9			Nimi: Merkki 5				
10			Nimi: Merkki 6				
11			Nimi: Merkki 7				
12			Nimi: Merkki 8				
13	3	3	End of text				
14			Ei käytössä				
15			Ei käytössä				
16			Vakio 1. tavu / Osoite 1. tavu	Luku 1.	KOMPASSISUUNTA		
17			Vakio 2. tavu / Osoite 2. tavu				
18			Vakio 3. tavu				
19			Vakio 4. tavu				
20			O=vakio Float type / 1=osoite	Luku 2.	Asetusarvo		
21			Vakio 1. tavu / Osoite 1. tavu				
22			Vakio 2. tavu / Osoite 2. tavu				
23			Vakio 3. tavu				
24			Vakio 4. tavu				
25			O=vakio Float type / 1=osoite	Luku 3.			
26			Vakio 1. tavu / Osoite 1. tavu				
27			Vakio 2. tavu / Osoite 2. tavu				
28			Vakio 3. tavu				
29			Vakio 4. tavu				
30			Ei käytössä				
31							
32							
33			Tulostus 1=on				
34							
35				Lähtö	Eroarvo		
36							
37							
38							
39			Hälytys päällä (min TAI max)				
40	%	37	Tietueen lopetus				
			= ei tarvitse syöttää mitään				

Kuvio 21. Kompassi-tomilohko Plazic-sovellusohjelmassa.

Autopilotin rakentaminen on vielä kesken. Opinnäytetyösuunnitelmassa tämä oli tiedostettu, mutta se ei estä vertailemasta toteutusta toisiin laitteisiin. Toimilaitteen paikoituksen toiminta on kokeiltu, ja kompassianturi on toiminnassa. Laitteen mekaniikka tarvitsee rakentaa loppuun ja tehdä Plazic-sovellusohjelma valmiiksi. Tutkimusaineistona käytettiin projektin suunnitteludokumentteja ja laitetoimittajien dokumentaatiota. Suunnitteludokumentit päivittyvät vielä, mutta vertailua varten dokumentaatio on riittävä. Veneilykausi on vasta aluillaan, ja koeajot päästään suorittamaan vasta myöhemmin kesällä.

Jos edellä kuvattua Plazic-sovelluksella tehtyä autopilottia verrataan Siemens S7 1200 logiikalla tehtyyn toteutukseen, tulee ratkaistavaksi magneettisen kulkusuunnan mittaaminen. Helpoin tapa on hankkia NMEA-väyläliitäntäinen venekäyttöön tarkoitettu kompassi. NMEA väylän voi liittää Profinet-väylään erillisellä muuntimella, esimerkiksi laitteella HD67459-E4V Industrial PROFINET to NMEA 2000 converter. Siemens-logiikoissa on Profinet-väylä käytössä, joten tieto

saadaan siirrettyä kompassilta logiikalle. Muu ohjelmointi ja moottorin ohjaaminen onkin tavanomaista logiikalla tehtävää ohjausta. Toimilaitteena voi käyttää kaupallista lineaarimoottoria.

Tyypillinen valmis pinnapilotti, mihin vertailua voisi suorittaa on Simrad TP10 automaattiohjain. Lineaarimoottorin yhteyteen on integroitu ohjaimen elektroniikka, kompassianturi ja painonapit. Laite tarvitsee toimiakseen vain 12V käyttöjännitteen. Simrad-pinnapilotti voidaan asentaa myös paapuurin puolelle, jolloin toimisuunta pitää kääntää. Laitteen viritysparametreista löytyy arvot säätimen vahvistukselle. Purjevenekäytössä energiankulutuksella on iso merkitys, voidaan määritellä ohjaukselle deadband eli kuollut alue. Säädin ei lähde korjaamaan virhettä, jos eroarvo on pienempi kuin kuollut alue. Toimilaitteen tekemät korjausliikkeet vähenevät ja sähköenergiaa kuluu vähemmän.



Kuvio 22. Simrad TP10 –ohjain (TP10, TP22, TP32 User guide n.d.)

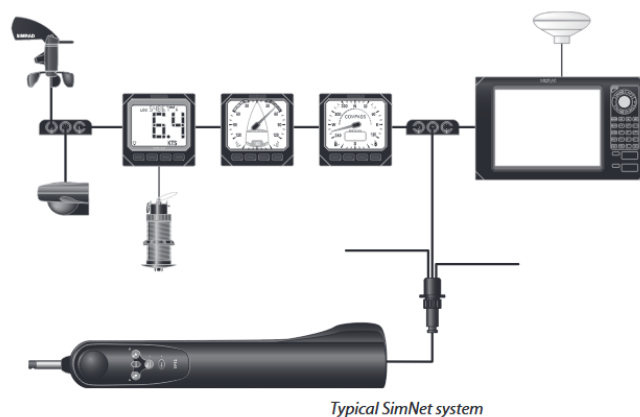
### 5.3 Arviointi

Laitteiden ominaisuuksia vertaillaan kappaleessa 3.3 määriteltujen arviointikriteerien mukaan. Kaikki ohjaimet soveltuvat teknisesti erittäin hyvin ohjaamaan alusta. Siemensin arviointia laskee laitteiston monimutkaisuus. Venekäytössä tarvitaan kompassiyksikkö, NMEA-Profinet-muunnin, 12 / 24 VDC jännitteenkorotusyksikkö ja logiikkayksikkö. Toimilaitteena olisi helpoin käyttää valmiita kaupallisia lineaarimoottoreita. Kaikki toiminnot on toteutettavissa, mutta toteutus on monimutkaisin (ks. taulukko 5).

Käyttäjäystävällisyys on loppukäyttäjän kannalta samanlainen, jos toteutus on vain painonapeilla. Plazic:n ja Siemensin vaatima ohjelmointi pudottaa luokituksen alemmaksi. Säätimen virittäminen veneolosuhteissa vaatii Siemensin logiikkatoteutuksessa tietokoneen käyttöä. Plazic:ssa sama onnistuu matkapuhelinta operoimalla.

Hinta nousee Siemensin logiikkatoteutuksessa korkeimmaksi. NMEA-Profinet muunnin maksaa n. 600 euroa. Kokonaishinta nousee reilusti yli 1000e. Simrad TP10 maksaa 469e. Arduinopohjaisen toteutuksen hinta jää pienemmäksi, vaikka ostaisi tehdasvalmisteisen lineaarimoottorin toimilaitteeksi. Ohjainyksikön kotelointi, kaapelointi ja muu käyttöönotto nostaa kuitenkin hintaa.

Muunneltavuus ja laajennettavuus on Siemensin S7 1200 logiikan parhaita puolia. Isommassa aluksessa saattaa olla paljon muutakin ohjattavaa. Uudemmissa isoissa aluksissa NMEA väylään liitetään lähes kaikki laitteet, moottorit, mittaristot, anturit jne. Muunneltavuus saattaakin olla joissain tilanteissa ohjaimen tärkein ominaisuus. Simrad TP10 on itsenäinen laite, jossa ei ole ulkoisia liitäntöjä. Laitteen laajennettavuutta ei siis ole. Samalta valmistajalta löytyy TP22 ja TP32 mallit, jotka voi kytkeä SimNet-väylään (ks. kuvio 23). Autopilottia voidaan ohjata väylän kautta.



Kuvio 23. SimNet väylä väylä (TP10, TP22, TP32 User guide n.d., 4)

Plazic:n muunneltavuus mahdollistaa erityyppisten ohjausten tekemisen toimilohko-ohjelmointina. Kuitenkaan Plazicssa ei ole ohjelmoituna väyläliitäntää NMEA-väylään.

GSM-viestiominaisuus saattaa kuitenkin olla monelle tarpeellinen. Varashälyttimen toiminta, tai pilssissä nouseva vesi voisi olla tärkeä tieto saada tekstiviestinä puhelimeen.

Taulukko 5. Autopilot-ohjaimien vertailu.

	Tekninen soveltuvuus	Käyttäjäystävällisyys	Hinta	Muunneltavuus
Siemens S7 1200	Hyvä	Hyvä	Huono	Erinomainen
Plazic	Erinomainen	Hyvä	Erinomainen	Hyvä
Simrad TP10	Erinomainen	Erinomainen	Hyvä	Huono

## 6 Tulokset

Kolmessa tutkitussa tapauksessa, dieselmoottorin pyörimisopeussäädössä, lämminvesivaraajan latauspumpun ohjauksessa ja veneen autopilotissa ohjausratkaisuja tutkittiin kolmella erityyppisellä ohjaimella. Arduino-pohjainen Plazic-sovellus oli arvioinnin kohteena ja Siemens S7 1200-logiikka oli vertailukohteena. Kolmanneksi vertailukohteeksi nostin jokaiseen toteutukseen tarkoitetun erillisen ohjausyksikön. Arviointiosuudessa nousee esille selkeästi Siemens S7 1200-laitteen ominaisuuksien monipuolisuus. Väyläliitännät ovat monipuolisia ja lisälaitteet mahdollistavat lähes kaikkiin laitteisiin liittymisen. Monimutkaisemmasta toteutuksesta tulee raskas. Logiikan pääasiallinen käyttö onkin teollisuusympäristössä, missä siitä ei ole haittaa. Plazic-sovellus on suunniteltu erityyppisiin tarpeisiin. Sovellusohjelmalla voi tehdä tavallisimmat ohjaukset. Toimilohkoja voi ohjelmoida helposti lisää. Väyläliitännöitä voi tehdä myös ja niihin löytyy valmiita ohjelmakirjastoja. SPI- ja I2C-väyliin voi liittää erilaisia sensoreita. GSM- ja Bluetooth-liitännät tuovat lisäedun laitteistolle. Plazic-sovelluksessa on selkeästi potentiaalia erityyppisten ohjausten toteuttamiselle. Avoimen lähdekoodin ansiosta räätälöinti on helppoa. Selkeäksi puutteeksi nousee graafisen HMI-näyttöyksikön puuttuminen. Suunnittelufilosofinen kysymys on, toteuttaako näytön erillisillä kosketusnäyttömoduuleilla vai tekeekö esimerkiksi Android-alustalle sovellusohjelman, joka kommunikoi Bluetoothin avulla Arduinon kanssa. Käyttäjäliittymä WWW-selaimella olisi laitteistovapaa tapa toteuttaa liitäntä. Myös useamman Arduinon yhteenliittäminen on tarpeen. Modbus tai Ethernet-pohjainen

kommunikointi mahdollistaisi paljon. Toivoisin Plazic-sovellusta käytettävän laajemminkin. Olen määritellyt ohjelmistolisenssin ilmaiseksi yksityiskäyttöön. Ohjelmiston kaupallisesta käytöstä toivoisin saavani vähän korvausta.

Hyvän näkökulman tutkittaviin tapauksiin toi valmiiden ohjausyksiköiden vertailu. Ohjausyksiköt ovat teknisiltä ominaisuuksiltaan sopivia ja edullisia laitteita. Jos ohjattavassa prosessissa on lisäksi muuta tarvetta automaatiolle, ei näitä laitteita kannata käyttää. Muuttaminen ja laajentaminen on mahdotonta, ainoastaan Simradin parempien mallien ominaisuuksissa oli väyläkommunikointimahdollisuus, joka mahdollistaa laajennettavuuden.

## 7 Pohdinta

Tavoitteena oli arvoida Plazic-sovelluksen ominaisuuksia vertailemalla sitä toisiin kaupallisiin ohjausjärjestelmiin. Työ määriteltiin ja rajattiin hyvin, ja lopputulokset ovat selkeitä. Jatkokehitykseenkin saatiin muutama selkeä kohde. Tutkimuksen arvioinnin luotettavuutta parantaa useamman tapauksen käsittely. Osa tutkimuksesta tehtiin arvioimalla valmistajan käyttöohjeista toteutusta. Tässä teoreettisessa arviointitavassa on omat riskinsä, koska vasta käytännön toteutuksessa nähdään todellinen onnistuminen. Plazic-sovelluksen arviointi on myös tehty suurimmaksi osaksi omasta toimestani ja näkökulma saattoi muodostua yksipuoliseksi. Käyttäjäkokemuksia olen saanut muilta käyttäjiltä esimerkiksi lämminvesivaraajan latauspumpun ohjauksesta. Plazic-sovelluksen luotettavuus on osoittautunut hyväksi. Ongelmat ovat olleet pääasiassa toimilaitteissa ja antureissa. Käytännön toteutuksissa olen käyttänyt Arduino Mega- alustaa. Kokeilu kaupallisiin Arduino-pohjaisiin logiikoihin on tekemättä. Myös standardien IEC 61131-1 ja -3 sisältöön olisi kiinnostanut perehtyä syvällisemmin ja arvioida toimintoja sen avulla.

Millaista automaation kehitys on tulevaisuudessa? Ohjausalgoritmit ovat vakiintuneet, mutta laitteistot kehittyvät. Keskitetyistä suurista järjestelmistä on ollut suuntaus hajauttaa I/O-liitynnät kentälle lähelle ohjattavia kohteita. Hajautetaanko ohjaustoiminnotkin tulevaisuudessa kentälle? Voisiko tulevaisuuden rakennusautomaatiossa olla esimerkiksi säädin-toiminto sisäänrakennettuna



toimilaitteeseen? Anturiliitännätkin tulisi suoraan toimilaitteelle, ja johdotustarve olisi minimaalinen. Väyläliityntä ja tehonsyöttö tulisi Ethernet PoE:lta. Uskoisin, että tulevaisuudessa on tarvetta Arduino-tyyppisille pienille automaatiolaitteille, kunhan sovellusohjelmien ja kokonaisuuksien yhteensovittaminen on helppoa.

Kotiautomaatioratkaisuissa on laitteiden yhteensovittaminen viety pitkälle.

Esimerkiksi Home Assistant- järjestelmässä on rakennettu laitteistorajapintoja yli tuhannelle erilaiselle laitteelle ja palvelulle. Teollisuusautomaation perinteinen lähestymisapa on käyttää standardoituja väyliä ja rajapintoja. Laitteiden välinen kommunikointi on pitänyt määritellä sovellusohjelmaan erikseen. Kehitystä tulee tapahtumaan varmasti tulevaisuudessakin. Tekniikka ei enää rajoita toteutustapoja, niin kuin automaation alkuaikoina tapahtui.

## Lähteet

- Atmel 8-bit microcontroller. 2011. Atmel. Viitattu 5.5.2019.  
<https://cdn.sparkfun.com/datasheets/Components/General%20IC/2549S.pdf>
- Compare board specs. N.d. Arduino Community. Viitattu 11.5.2019.  
<https://www.arduino.cc/en/products/compare>
- Digital Loggers PLDuino. N.d. Adafruit. Viitattu 11.5.2019.  
<https://www.adafruit.com/product/3418>
- ESD-5550/5570 Speed control unit manual. N.d. Huegli Tech. Viitattu 5.5.2019.  
[https://www.huegli-tech.com/fileadmin//Dateien/pdfs/products/governingsystems/SpeedGovernors/MN-Neu/ESD-5550-5570\\_MN\\_EN\\_07.2011.pdf](https://www.huegli-tech.com/fileadmin//Dateien/pdfs/products/governingsystems/SpeedGovernors/MN-Neu/ESD-5550-5570_MN_EN_07.2011.pdf)
- Heinisuo, O., 2012. Rakentelijan unelma. Artikkelilehdessä MikroPC, 10, 30-35.
- HPS-CB700 Standard surface servo. N.d. Futaba. Viitattu 5.5.2019.  
<https://futabausa.com/shop/servos/hps-cb700/>
- Ihalainen, J. N.d. Sulautettujen järjestelmien arkkitehtuurit. Luentomoniste Tietoliikennelaboratorio Kokkolan yliopistokeskus. Viitattu 5.5.2019.  
<http://users.jyu.fi/~jupeihal/Yleinen%20arkkitehtuuri.pdf>
- Kananen, J. 2013. Case-tutkimus opinnäytetyönä. Jyväskylä: Jyväskylän ammattikorkeakoulu. Jyväskylän ammattikorkeakoulun julkaisu.
- Keinänen, T. Kärkkäinen, P., Metso, T. & Putkonen, K. 2002. Logiikat ja ohjausjärjestelmät. Porvoo: WSOY.
- Kippo, K. & Tikka, A. 2008. Automaatiotekniikan perusteet. Helsinki: Edita
- Marine autopilotit. N.d. Garmin. Viitattu 11.5.2019. <https://buy.garmin.com/fi-FI/FI/cOnTheWater-cAutopilots2-p1.html>
- M-Duino PLC. N.d. Industrial Shields. Viitattu 11.5.2019.  
<https://www.industrialshields.com/shop/product/is-mduino-58-m-duino-plc-arduino-ethernet-58-i-os-analog-digital-plus-176>
- Pelagic Autopilot. 2017. My Boats Gear. Viitattu 11.5.2019.  
<http://myboatsgear.com/2017/08/03/pelagic-autopilot/>
- Sundquist, M., Logiikat: standardi IEC 61131-1 ja -3. Luentomoniste Sesko. Viitattu 11.5.2019. [https://www.sesko.fi/files/101/osio\\_9.pdf](https://www.sesko.fi/files/101/osio_9.pdf)

Temperature difference controller. N.d, käyttöohje Sorel STDC. Viitattu 5.5.2019. [https://www.nn-energy.se/sites/default/files/products/%2C%20node%2C%2033%2C%20edit/dokument/STDC\\_en.pdf](https://www.nn-energy.se/sites/default/files/products/%2C%20node%2C%2033%2C%20edit/dokument/STDC_en.pdf)

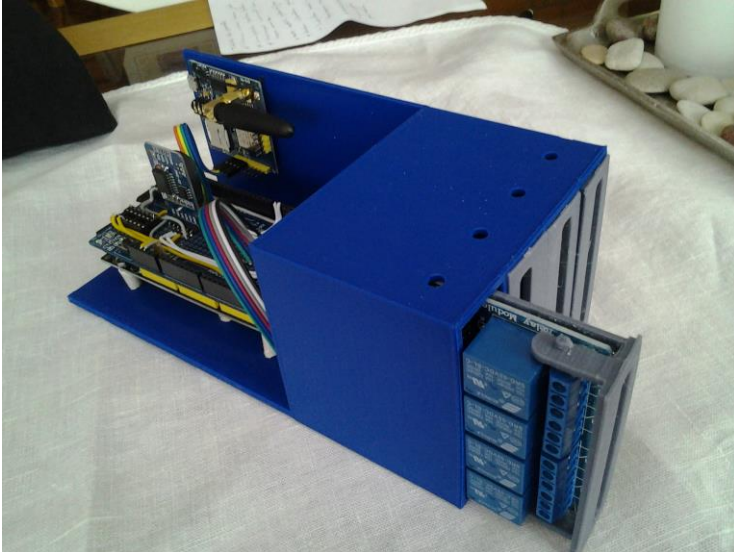
TP10, TP22, TP32 User guide. N.d. Simrad. Viitattu 12.5.2019. [https://ww2.simrad-yachting.com/Root/TillerPilots/TP10-22-32\\_OM\\_EN\\_988-10413-003\\_w.pdf](https://ww2.simrad-yachting.com/Root/TillerPilots/TP10-22-32_OM_EN_988-10413-003_w.pdf)

## **Liitteet**

Liite 1. Plazic-sovelluksen käyttöohje

ver3.0 beta

16.5.2019



## Käyttöohje Plazic-ohjelmalle

Ohjelma on tehty yleiskäyttöiseksi Arduinon ohjelmaksi, joka pystyy hoitamaan tavallisimmat ohjelmoitavan logiikan toiminnot. Myös säädintoiminnot on versiossa Beta 2.0 alkaen. Ohjelmalla voi myös tehdä tekstiviestihälytykset ja ohjaukset. Ohjelmassa on myös analogiasignaalien käsittely (mm lämpötilamittaus, raja-arvohälytys ym). Parasta Plazic-ohjelmassa on se, että voit käynnin aikana muuttaa ohjelmaa tai parametreja!!! Ohjelman lisenssiehdot liitteenä. Ohjelmaa saa käyttää vapaasti ei kaupalliseen käyttöön. Naapurille ja kaverille voi tietenkin jotain tehdä myös. Jos sovelluksen kaupallinen käyttö kiinnostaa, ota yhteyttä ohjelmiston tekijään (yhteystiedot lisenssi liitteessä). Ohjelman tekijä ei ota mitään vastuuta ohjelman aiheuttamista vahingoista.

Jos tuntui, että ohjelmistosta on sinulle hyötyä, voit laittaa hyvän kiertämään. Lahjoita esimerkiksi joku roponen kehitysapuun: <https://www.fida.info/lahjoita/>

PLazic- sovellus on Arduinoon ladattava ohjelma, jolla voi tehdä tavallisimmat ohjaukset ja välittää hälytykset matkapuhelimeen. Jos Arduinoon on asennettu:

Reaaliaikakello :        Voit tehdä kello-ohjaukset (Tiettyyn kellon aikaan päälle/pois ohjaus, tai mikä tahansa ohjaus)

GSM Modeemi            - Hälytys tekstiviestillä puhelimeen  
                              - Päälle / pois ohjaus tekstiviestillä (vapaasti määriteltävissä)

Ohjelman käyttäjän ei tarvitse osata Arduinon ohjelmointikieltä. Toimintojen määrittely tapahtuu matkapuhelimen Bluetoothiin, tabletin tai tietokoneen sarjamonitorin kautta yksinkertaisilla käskyillä. Mitään erillisiä sovelluksia ei välttämättä tarvita, vaan tavallisilla sarjaliikenne sovelluksilla pärjätään.

Laitevaatimukset:    - Arduino MEGA (tai muu EPROM:lla varustettu Arduino ks. Ohje1 liitteenä)  
                              - Reaaliaikakello RTC (ei pakko, jos ei tee kello-ohjauksia tai viiveitä)  
                              - A6 GSM Modeemi + SIM kortti (ei pakko, jos ei halua tekstiviestitoimintoja)  
                              - Bluetooth modeemi (Jos haluaa puhelimella tai tabletilla ohjata)  
                              - Relekortti ohjauksia varten  
                              - ym Arduino laitteet, lämpöanturit, servot....  
                              -

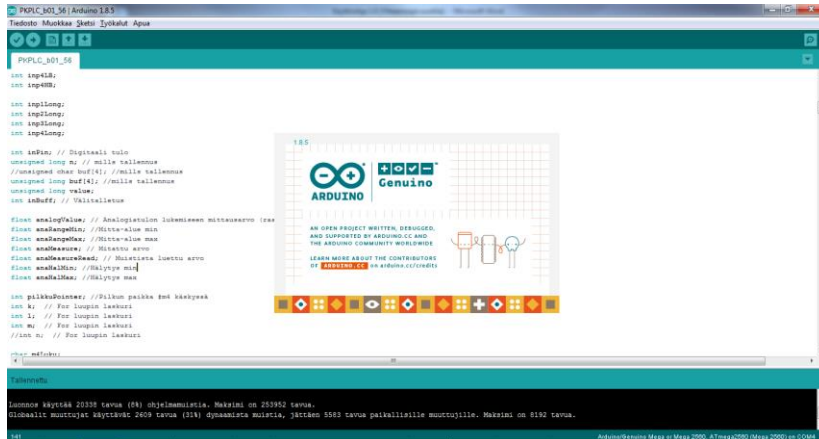
Käyttöohjeessa ei ole kytkentäohjeita. Netistä löytyy ohjeita suomeksikin, esim:

<https://www.hutasu.net/mikrokontrollerit/arduino/>

Reaaliaikakello on kytketty SPI-väylään. GSM modeemi Serial 3:een (Arduino MEGA).

## Ohjelman asentaminen tietokoneelle

Tarvitset tietokoneellesi Arduini IDE:n (Arduinon ohjelmoimiseen ja ohjelmien lataamiseen tarkoitettu ohjelma). Kuva: Arduino IDE ikkuna tietokoneella.



Ohjelman asentamiseen löytyy netistä paljon ohjeita. Lue ja asenna esim:

<https://github.com/Robojuchen/Mechatronics2013/wiki/Alkuun-Arduinolla!>

Tai

(Kohta 2. Arduino IDE:n asennus)

<https://www.avi.fi/documents/10191/7858814/Ohjelmien+asennusohjeet.pdf/e5d5299f-2d74-4a25-b14a-5d57dea357b1>

## Lataa ohjelma

- Lataa Plazic-ohjelma netistä kovalevylläsi.
- Aavaa Arduino IDE:en Plazic ohjelma, ja lähetä se Arduinoosi.

## Kirjastot

Ohjelma käyttää joitakin kirjastoja. Koneellesi täytyy olla asennettuna:

<EEPROM.h> <DS3231.h> <Wire.h> <Servo.h> <SoftwareSerial.h> kirjastot.

## Yksinkertaisen ohjelman tekeminen

Hälytystulo:

Jos sinulla on uusi Arduino Mega. Eeprom (=haitumaton muisti) on tyhjä, ja sinne täytyy tehdä ohjelma. Yleisimpien toimilohkojen tekemiseen löytyy valmis komento. Muuten joudut syöttämään rivi kerrallaan numerot, tai kopioimaan olemassa olevaa.

Muistiin syötetään sisältö-sarakkeen 30 numeroa (voit jättää syöttämättä 15-18 ja 20-23 rivit, ja nimi- rivit 5-12). Muistitilaa on n 1000 riviä, ja sinun pitää päättää, mihin ohjelma syötetään. Hyvä on aloittaa vaikka muistipaikasta 100. Komento #crdi tekee allaolevan automaattisesti.

Rivin 27 sisältö pitää olla sama, mihin kytkimen johdot on kytketty. (Esimerkin liitin on 45, joka on myös kanavanumero.)

### Kuva: Muistipaikkojen merkitys

INPUT			
Digitaalitulo			
Muistipaikka	Sisältö		
	1	35	#
	2	30	30
	3	98	b
	4	1	1
	5	84	T
	6	117	u
	7	108	l
	8	111	o
	9	49	1
	10	3	3
	11	0	
	12	0	
	13	3	3
	14	0	
	15	0	
	16	0	
	17	0	
	18	0	
	19	0	
	20	0	
	21	0	
	22	0	
	23	0	
	24	0	0
	25	1	0
	26	0	
	27	45	
	28	0	
	29	0	
	30	37	%

Tietueen aloitus

Tietueen pituus

binääritulo

Versio

Nimi: Merkki 1

Nimi: Merkki 2

Nimi: Merkki 3

Nimi: Merkki 4

Nimi: Merkki 5

Nimi: Merkki 6

Nimi: Merkki 7

Nimi: Merkki 8

End of text

Lukuviive 0= sek, 1=min

0=kummatkin viiveet, 1= vetohidastus, 2=päästöhidastus

0= normaali toiminta, 1=Pehmopiste (tila luetaan osoitteesta)

Pehmopisteen osoite

Tekstiviestihälytys = 1, ei hälytystä =0

Aikalaskurin lukema

lukuviive 0-255 sek/min (ks muistipaikka 14)

Hälytys tulostus 0=ei, 1=päällä/pois, 2=hälytys/hälytys poistui

Edellinen tila (nousevan reunan tutkiminen)

Pin numero (kanava). Liitin, mihin olet kytkenyt tiedon.

Invertointi 0=ei, 1=invertoidaan tolo

Lähtö 0/1 (kaikki yli 1 on 1)

Tietueen lopetus

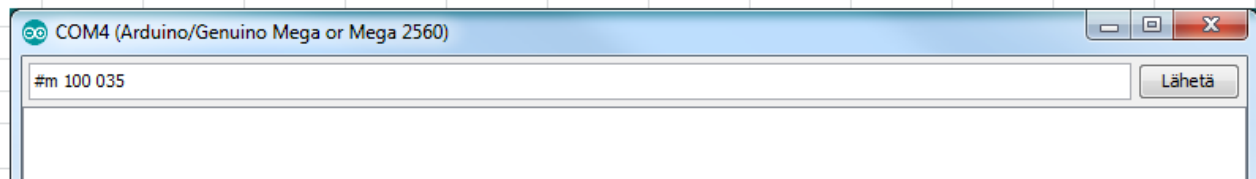
= ei tarvitse syöttää mitään

Hyvä on aloittaa vaikka muistipaikasta 100.

INPUT				Digitaalitulo		Kirjoita komentoriville	
Muistipaikka		Sisältö					
100	1	35	#			#m 100 035	
101	2	30	30			#m 101 030	
102	3	98	b			#m 102 098	
103	4	1	1				
104	5	84	T				
105	6	117	u				
106	7	108	I				
107	8	111	o				
108	9	49	1				
109	10	3	3				
110	11	0					
111	12	0					
112	13	3	3				
113	14	0					
114	15	0					
115	16	0					
116	17	0					
117	18	0					
118	19	0					
119	20	0					
120	21	0					
121	22	0					
122	23	0					
123	24	0	0				
124	25	1	0				
125	26	0					
126	27	45					
127	28	0					
128	29	0					
129	30	37	%				

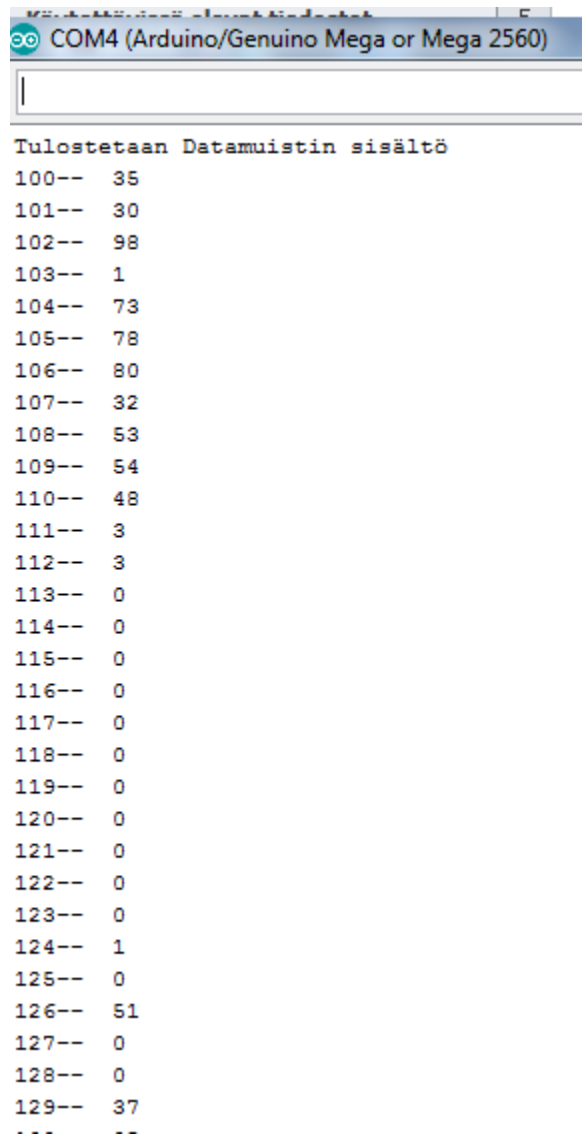
Eli ensimmäinen rivi syötetään näin: **#m 100 035** (jos teit #crdi komennolla, ei rivejä tarvitse syöttää. Muuta vain kanava #m 126 xxx (jossa xxx on kanavanumero esim 051)





Kirjoita kaikki rivit huolella (huom välilyönnit, ja syötä aina kolme numeroa, nolla on siis 000!!!)

Voit nyt listata syöttämäsi arvot. Kirjoita komentoriville käsky **#pr all** (print memory)



kuva: Ohjelman tulisi näyttää suurin piirtein tältä. 126 muistipaikassa on käyttämäsi kanava(liitin, mihin johdon olet kytkenyt)

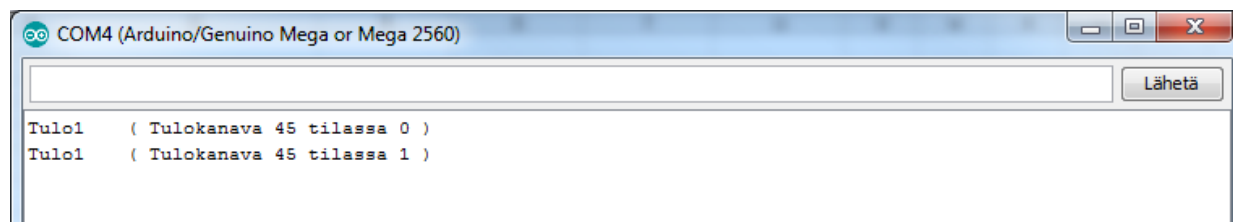
Ohjelma on jo käynyt suorittamassa, ja muuttamasa joitakin muistipaikkoja!! Tyhjä muisti on täynnä 255-numeroita (muistiin voi syöttää vain luvun 0-255!!!).

Kirjoita nyt käsky **#mw** (memory write), jolla tallennetaan ohjelma haihtumattomaan EPROM-muistiin.

Ohjelma tekee käynnistyessään vain kanaviin liittyvät alustukset. (kopioidessa tulisi muuten päällekkäisiä määrittelyitä!!). Paina reset-nappia Arduinostasi, niin päästään testaamaan!!!

Ohjelman testausta:

Jos kaikki on mennyt hyvin, ja muutat nastan 45 tilaa, sarjaliikennemonitoriin tulostuu:



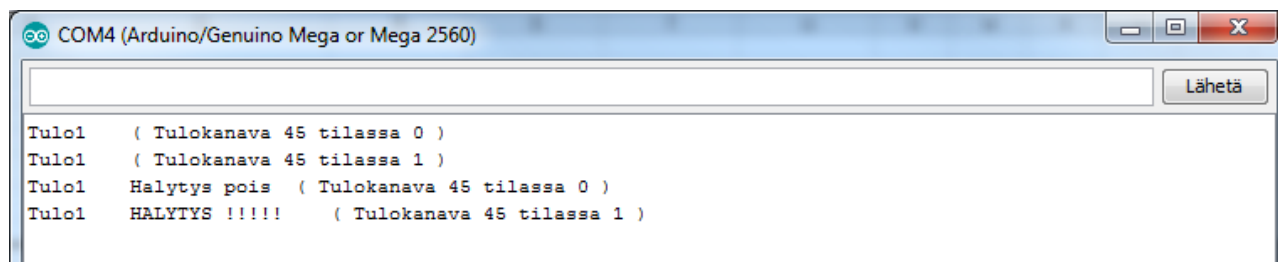
Jos haluat muuttaa tulon hälytystuloksi

124

Hälytys tulostus 0=ei, 1=päällä/pois,2=hälytys/hälytys poistui

Muuta muistipaikka 124 nolasta kakkoseksi:

Kirjoita nyt käsky **#m 124 002** (m=memory),



Voit muuttaa:

- 127 muistipaikalla kytkimen toiminnon käänteiseksi (invertointi)
- 118 muistipaikalla tekstiviestihälytyksen päälle (pitää määritellä modeemin asetukset ensin)
- 123 muistipaikalla viiveaika, jonka jälkeen kytkintieto vaihtaa tilaa (sekunteja)

Muista tallettaa asetukset #mw käskyllä!!!

Muita hyviä käskyjä:

Jos haluat poistaa toimilohkon käytöstä, riittää, kun kirjoitat ensimmäiseen muistipaikkaan nolla. Ohjelma ei lue toimilohkoa sen jälkeen.

**#lb** (list blocks) listaa kaikki muistiin syöttämäsi toimilohkot

Tulostetaan muistista löytyvät toimilohkot. Lisää tietoa toimilohkon sisällöstä komennolla #in xxx (xxx on lohkon osoite)

GSM ohjaus	muistipaikassa 100 Nimi: Valo on Tila: 0
RS toimilohko	muistipaikassa 150 Nimi: RS-lohko Tila: 0
GSM ohjaus	muistipaikassa 170 Nimi: Valo off Tila: 0
DO Digitaalilähtö	muistipaikassa 230 Nimi: Lahto1 Tila: 1 Ohjaus osoitteesta 12
DO Digitaalilähtö	muistipaikassa 250 Nimi: Lahto2 Tila: 0 Ohjaus osoitteesta 475
DO Digitaalilähtö	muistipaikassa 270 Nimi: Lahto3 Tila: 0 Ohjaus osoitteesta 476
DO Digitaalilähtö	muistipaikassa 300 Nimi: Lahto4 Tila: 0 Ohjaus osoitteesta 168
GSM Numero	muistipaikassa 320 Nimi: +358401955337 Tila: 255
DI Digitaalitulo	muistipaikassa 350 Nimi: Palo Tila: 1
DO Digitaalilähtö	muistipaikassa 400 Nimi: Led01 Tila: 0 Ohjaus osoitteesta 448
DI Digitaalitulo	muistipaikassa 420 Nimi: Led_DI Tila: 0
AI Analogiatulo	muistipaikassa 450 Nimi: Paine Tila: 255
DI Digitaalitulo	muistipaikassa 500 Nimi: DI_50 Tila: 0
DI Digitaalitulo	muistipaikassa 530 Nimi: DI_5 Tila: 0
DI Digitaalitulo	muistipaikassa 560 Nimi: DI_52 Tila: 0
DI Digitaalitulo	muistipaikassa 590 Nimi: DI_53 Tila: 0

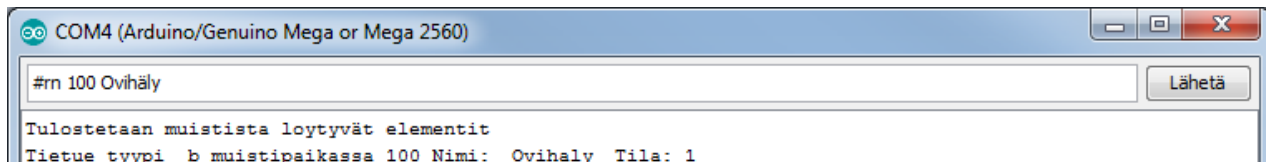
Listalla on kaikki muistissa olevat toimilohkot, ja digitaalisten toimilohkojen tilat. Jos haluat lisätietoa esimerkiksi alimmasta lohkoista, kirjoita #in 590 (info-komento).

**#in xxx** (info) Kertoo toimilohkon sisällön ja komennon, millä voi muokata. Toimii vain yleisimmille toimilohkoille (DI,DO,AI)

```
-----D I G I T A A L I      T U L O -----  
Ver 1  
Käyttää muistialuetta: 590 - 619  
DI_53 Muokkaa nimeä komennolla #rn 590 MAX 8 merkkia!!!!  
  
Tekstiviesti:      0  (0=ei 1=lähetää tekstiviestin) Muuta komennolla #m 608  
Lukuviive:         2  (0-255 sek) Muuta komennolla #m 613  
Hälytys tulostus:  1  (0=ei, 1=päällä/pois,2=hälytys/hälytys poistui ) Muuta komennolla #m 614  
Kanava/Liitin nro: 52 (Piirilevyllä numero liittimen vieressä) Muuta komennolla #m 616  
Invertointi:       0  (0=normaali toiminta, 1=kaanteinen) Muuta komennolla #m 617  
Tulon tila         0  Tätä paikkaa ei voi muokata, mutta voit kytkeä toisen toimilohkon tähän muistipaikkaan nro: 618  
----- Muista tallettaa muutokset!!! -----
```

Ikkunassa näkyy käsky, millä esimerkiksi tekstiviestihälytyksen saa päälle.

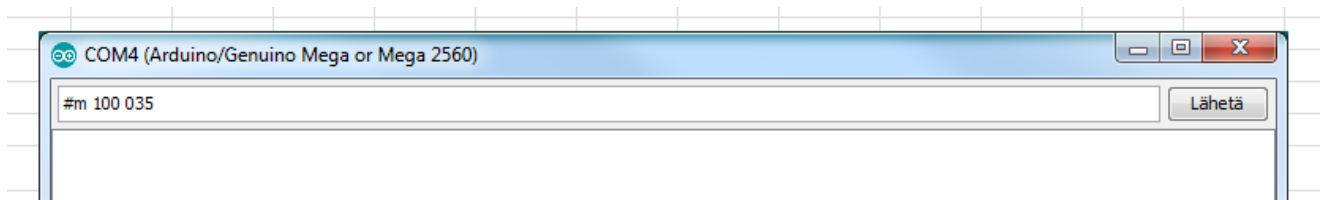
**#rn xxx uusinimi** (HUOM Max 8 merkkiä!!!) (ReName) nimeää elementin uudelleen



(Saat muutokset näkyviin #lb käskyllä)

Nimen maksimipituus on 8 merkkiä. Ohjelmassa ei ole estoa ylipitkän nimen kirjoittamiselle. Jos kirjoitat liian pitkän nimen, toimilohkosi muistialue sotkeutuu. (Aina voit palauttaa #mr komennolla EEPROM:sta sinne tallennetun, jos huomaat virheen ennen tallennusta).

Eli ensimmäinen rivi syötetään näin: **#m 100 035**



Kirjoita kaikki rivit huolella (huom välilyönnit, ja syötä aina kolme numeroa, nolla on siis 000!!!)

**#ml xxx yyy** (memory long write), jossa xxx on muistipaikka 1-999 ja yyy on data.

Esimerkiksi kuvan OUTPUTIN muisti on kirjoitettu

**#ml 154 128**

Muistin sisältö voisi olla esimerkiksi 260 jolloin ohjelma kirjoittaisi paikkoihin 154 ja 155 luvut joista ei saisi selvää ilman laskemista.

## Tiedon siirtäminen ohjelman sisällä

Kaikkien muistissa olevien toimilohkojen välillä voi siirtää tietoa(kytkeä toimilohkoja toisiinsa). Toimilohkot lukevat toisiltaan tietoa seuraavien periaatteiden mukaan:

INPUT Digitaalitulo				
Muistipaikka	Sisäto			
100	1	35	#	Tietueen aloitus
101	2	30	30	Tietueen pituus
102	3	98	b	binääritulo
103	4	1	1	Versio
104	5	94	T	Nimi Merkki 1
105	6	117	u	Nimi Merkki 2
106	7	108	I	Nimi Merkki 3
107	8	111	o	Nimi Merkki 4
108	9	49	1	Nimi Merkki 5
109	10	3	3	Nimi Merkki 6
110	11			Nimi Merkki 7
111	12			Nimi Merkki 8
112	13	3	3	End of text
113	14			
114	15			
115	16			
116	17			
117	18			
118	19			Tekstivälilyhyys = 1, ei hälytystä = 0
119	20			Aikakulun lukema
120	21			
121	22			
122	23			
123	24	0	0	Lukuväli 0-255 sek
124	25			Hälytys tulostus 0=ei, 1=paikallpois, 2=hälytys/hälytys poimittu
125	26			Edellisen tilin (noustaan reuna) yllä
126	27	45		Pin numero 0-255 sek
127	28	0		Invertointi 0=ei, 1=invertoidaan tolo
128	29			Lähtö 0/1 (kaikki yli 1 on 1)
129	30	37	%	Tietueen lopetus

OUTPUT Digitaalilähtö				
150	1	35	#	Tietueen aloitus
151	2	20	20	Tietueen pituus
151	3	100	d	binääritulo
152	4	1	1	Versio
152	5	78	L	Nimi Merkki 1
151	6	97	a	Nimi Merkki 2
152	7	104	h	Nimi Merkki 3
152	8	116	t	Nimi Merkki 4
153	9	111	o	Nimi Merkki 5
153	10	49	1	Nimi Merkki 6
152	11	3		Nimi Merkki 7
153	12			Nimi Merkki 8
153	13	3	3	End of text
154	14			
154	15	22		Pin numero (kanava)
153	16	0		Invertointi 0=ei, 1=invertoidaan tolo
154	17	128		Lähtö ohjaus (osoite, mistä luetaan)
154	18			
155	19			Lähdön tila
155	20	%	37	Tietueen lopetus

Lohkojen kytkennän periaate: Jos tietoa siirretään lohkojen välillä, käy tietoa haluava lukemassa tiedon muistipaikasta. Tässä kuvassa Output hakee tiedon Input toimilohkolta muistipaikasta 128. Eli tulo ohjaa suoraan lähdön päälle.

Tiedon siirtäminen haluttiin tehdä luku-periaatteella, ja minimoida toisen toimilohkon muistialueen muokkaaminen. Muistiin voi kirjoittaa vain lukuarvoja 0-255. Kuitenkin meillä on n 1000 merkkiä (vieläkin enemmän, mutta toistaiseksi max nro on 999). Ongelma on ratkaistu käyttämässä kahden muistipaikan yhdistettyä sisältöä. Toimilohkojen lähtötieto on yleensä aina toiseksi viimeisessä muistipaikassa. Digitaalitulon tieto on luettavissa riviltä 29, eli muistipaikasta 128.

## Esimerkki kello-ohjauksesta

Kello ohjauksessa tarvitaan kaksi toimilohkoa:

KELLO-OHJAUS					
Muistipaikka		Sisältö			
150	1	35	#	Tietueen aloitus	
151	2	20	20	Tietueen pituus	
152	3	67	C	Kello-ohjaus	
153	4	1	1	Versio	
154	5	76	L	Nimi: Merkki 1	
155	6	97	a	Nimi: Merkki 2	
156	7	104	h	Nimi: Merkki 3	
157	8	116	t	Nimi: Merkki 4	
158	9	111	o	Nimi: Merkki 5	
159	10	49	1	Nimi: Merkki 6	
160	11	3		Nimi: Merkki 7	
161	12			Nimi: Merkki 8	
162	13	3	3	End of text	
163	14			Päälle klo: Tunnit	
164	15			Päälle klo: Minuutit	
165	16			Pois klo: Tunnit	
166	17			Pois klo: Minuutit	
167	18			Varaus	
168	19			Lähdön tila	
169	20	37	%	Tietueen lopetus	

OUTPUT					
Digitaalilähtö					
Muistipaikka		Sisältö			
170	1	35	#	Tietueen aloitus	
171	2	20	20	Tietueen pituus	
172	3	100	d	binääritulo	
173	4	1	1	Versio	
174	5	76	L	Nimi: Merkki 1	
175	6	97	a	Nimi: Merkki 2	
176	7	104	h	Nimi: Merkki 3	
177	8	116	t	Nimi: Merkki 4	
178	9	111	o	Nimi: Merkki 5	
179	10	49	1	Nimi: Merkki 6	
180	11	3		Nimi: Merkki 7	
181	12			Nimi: Merkki 8	
182	13	3	3	End of text	
183	14				
184	15	22		Pin numero (kanava)	
185	16	0		Invertointi 0=ei, 1=invertoidaan tolo	
186	17	168		Lähtö ohjaus (osoite, mistä luetaan)	
187	18	0			
188	19			Lähdön tila	
189	20	37	%	Tietueen lopetus	

Kello-ohjaus toimilohkon lähtö 19 (=esimerkin osoite 168) menee tilaan ”1”, kun kellon aika on päälle ja pois aikojen välissä. Muuloin lähtö on ”0”. Digitaalilähtö lukee 17-18. rivillä (=esimerkin osoite 186) osoitteesta 168 ohjauksen, eli lähtöliitin 22 ohjautuu päälle. Samaa kellon lähtöä voi lukea useat toimilohkot samanaikaisesti.

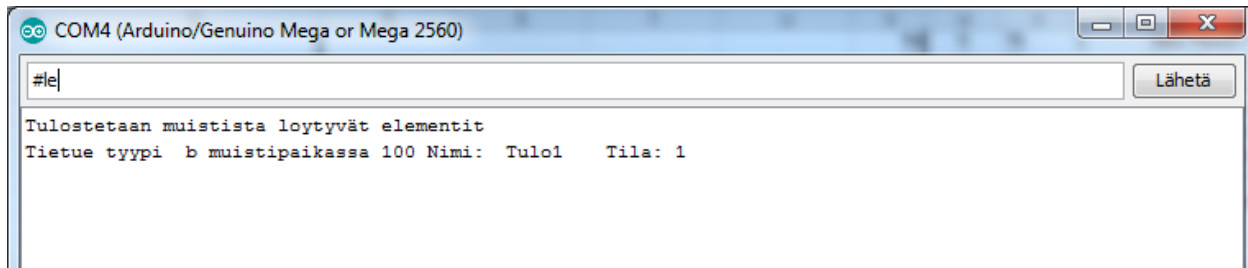
Lähdön ohjaus osoite on riveillä 17-18 (kaksi muistipaikkaa). Yhteen muistipaikkaan voi kirjoittaa vain lukuarvon 0-255. Käytä aina muokkaamiseen komentoa #ml, niin komento tekee kumpaakin muistipaikkaan oikean luvun. (esim #ml 186 500). Voit tarkistaa kirjoittamisesi #pl 500 (tulostaa siis muistipaikan 500 ja 501 paikoista lasketun datan).

Kello pitää olla asetettuna aikaan #st komennolla (set time). Aseta ajat muistipaikkoihin 14-17!!!

## Komentoriville annettavat komennot

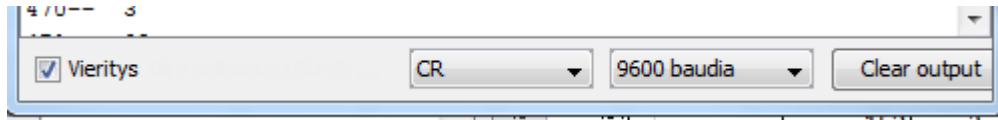
Voit antaa komentoja tietokoneella, Arduinon ohjelmointityökalun Sarjamonitori-ikkunasta tai matkapuhelimen sarjaliikennesovelluksesta Bluetoothin kautta. Huomaa, että bluetooth käyttää samaa USB liittimen sarjaliikenneporttia, eli kumpaakin ei voi käyttää yhtä aikaa!!!

Kuva: Komennon #lb kirjoittaminen:



Komento alkaa aina #-merkillä. Komento päättyy Enter-painallukseen.

Tarkista, että ikkunassa on "CR" valinta päällä.



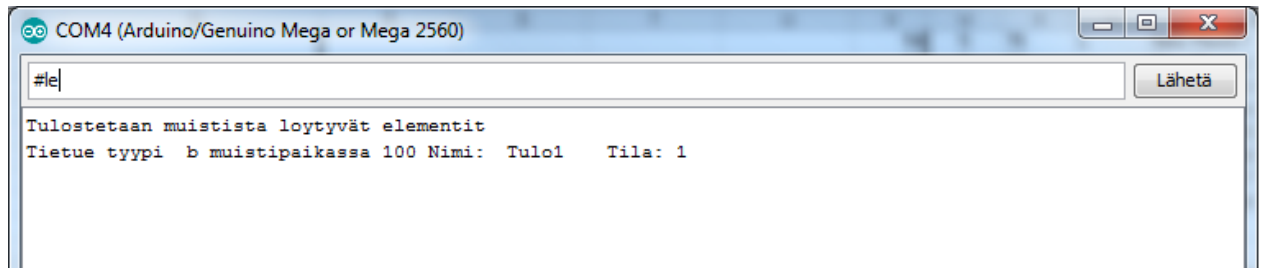
Komentojen numerot annetaan aina kolmella numerolla. Nolla on siis 000.

Ohjeen alussa olevassa esimerkissä on käyty komentojen käyttämiseen liittyvät esimerkit läpi.

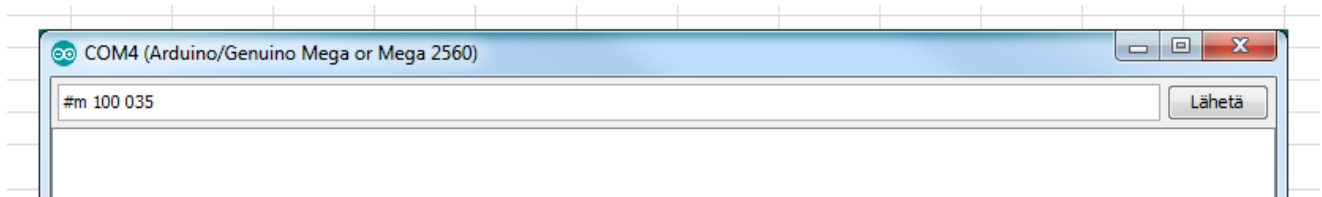


**Kopiointi:** **#c xxx yyy zzz** (copy) Kopioi muistipaikasta x -> muistipaikkaan y z määrän soluja.  
Esim **#c 100 200 030** (kopioi solut 100-130 -> 200 – 230)

**Listaus:** **#lb** (list blocks) listaa kaikki muistiin syöttämäsi toimilohkot

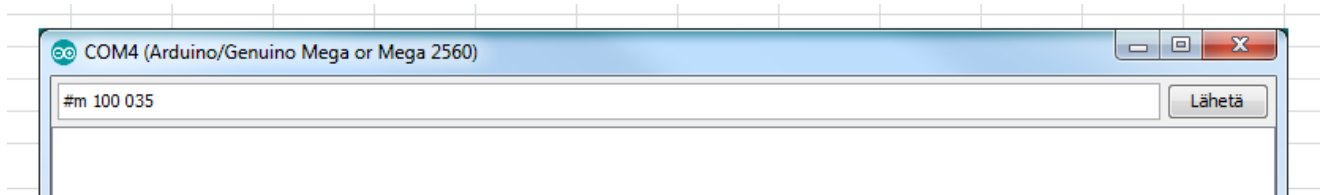


**Muistipaikkaan syöttö:** **#m xxx yyy** (memory) Syöttää x muistipaikan arvoksi luvun yyy



Kirjoita kaikki rivit huolella (huom välilyönnit, ja syötä aina kolme numeroa, nolla on siis 000!!!). Muistipaikan arvo yyy saa olla enintään 255!!!!

**Pitkä Muistipaikkaan syöttö:** **#ml xxx yyy** (memory) Syöttää x muistipaikan arvoksi luvun yyy. Arvo yyy saa olla 0 – 999.

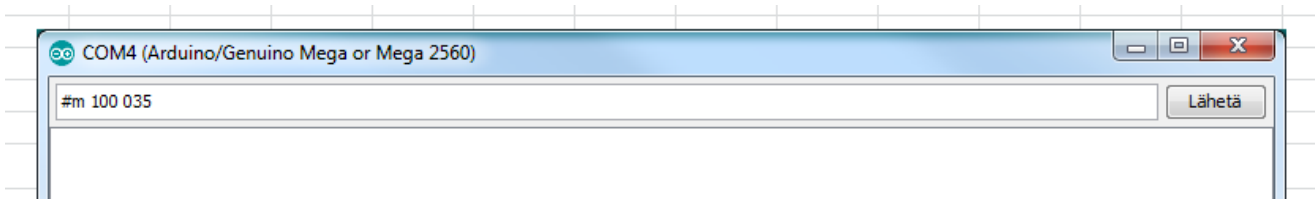


Kirjoita kaikki rivit huolella (huom välilyönnit, ja syötä aina kolme numeroa, nolla on siis 000!!!)

Käskey kirjoittaa kahteen peräkkäiseen paikkaan. Varmistu, missä käskey voi käyttää (toimilohkojen tiedot).

**Pitkä Muistipaikkaan syöttö:** **#m4 xxx yyy.yyy** (memory) Syöttää x muistipaikan

arvoksi luvun yyy. Arvo yyy saa olla desimaaliluku muotoa 000.00



Kirjoita kaikki rivit huolella (huom välilyönnit, ja syötä aina kolme numeroa, nolla on siis 000!!!)

Käskey kirjoittaa neljään peräkkäiseen paikkaan (liukukuku). Varmistu, missä käskey voi käyttää (toimilohkojen tiedot). Syötä aina piste ja joku luku sen jälkeen. Esim 100 -> 100.0. (huonoa koodia ☹) Voit katsoa sisällön #p4 käskeyllä.

## EPROM muisti työmuistiksi

**#mr** (memory read) kopioi haitumattomasta muistista tiedot työmuistiin. Käskey alustaa myös digitaalitulot ja lähdöt. Arduinon käynnistyessä ohjelma lukee automaattisesti EPROM muistin työmuistiin.

## Työmuistin tallennus EPROM:iin

### #mw (memory write)

Huom!! Kaikki lähdöt ym tallettavat myös tähän tilaan. Uudelleen käynnistyessä lähtötilanne on tämä!!!

## Muistipaikan tulostus

**#p4** (Print 4 byte) tulostaa #m4 käskeyllä tallennnetun muistipaikan sisällön näytölle (tietotyyppi: liukuluku , esim lämpötilamittaus)

## Muistipaikan tulostus

**#pm** (Print 1 byte) tulostaa #m käskeyllä tallennnetun muistipaikan sisällön näytölle (tietotyyppi: tavu, luku voi olla välillä 0-155)

## Muistipaikan tulostus

**#pl** (Print 2 byte) tulostaa #ml käskeyllä tallennnetun muistipaikan sisällön näytölle (tietotyyppi 16 bit kokonaisluku, kaksi tavua, 0...65535)

## Muistialueen tulostus

**#pr all** (print) tulostaa näytölle koko muistialueen

**#pr xxx** Jossa xxx on muistipaikka. Tulostaa seutaavat 50 paikkaa

## Kellon ajan tulostus

**#pt** (print time) tulostaa näytölle kellon ajan

## Kellon ajan asetus

**#st** (set time) asettaa kellon aikaan (Syntaksi #st tt:mm)

PUUTTUU  
vielä.....

#sd Set Date

Päiväyksen asetus

#sw Set date of Week

Viikonpäivän  
asetus

## AT-komento modeemille

**#AT XXXXX** Lähettää modeemille koko komennon ilman # merkkiä.

## Toimilohkon nimeäminen

**#rn XXX ABCDEFGH** (ReName) nimeää toimilohkon. Max 8 merkkiä.

Laita xxx osoitteeksi ensimmäinen toimilohkon osoite. Sen saat helpoiten #lb komennolla

(GSM tekstiviestin numero syötetään samalla käskyllä, voi olla 20 merkkiä pitkä)

## Luo digitaalitulo

**#crdi** Luo digitaalitulon tyhjään muistialueeseen. Muuta kanava ja nimi.

## Luo digitaali lähtö

**#crdo** Luo digitaali lähdön tyhjään muistialueeseen. Muuta kanava ja nimi.

## Luo analogiatulo

**#crai** Luo analogiatulon tyhjään muistialueeseen. Muuta kanava ja nimi, mitta-alueet ja hälytysrajat.

## Tyhjennä muisti

**#clrm** Tyhjentää sovellusohjelman RAM-muistista. Jos haluat tyhjentää EPROM:n, tallenna #mw-komennolla tyhjä muisti

## Ohjelman pysäytys

**#stop** Pysäyttää ohjelman suorittamisen. Jos lähetät

sarjaliikenne-ohjelmalla varmuuskopion, pysäytä ohjelma ensin.

## Ohjelman käynnistys

**#start** Käynnistää ohjelman suorittamisen.

### Ohjelman varmuuskopiointi

Voit vastaanottaa tiedostoja paremmilla sarjaliikennepääteohjelmilla. Tallenna muistin sisältö #pr all –komennolla.

### Ohjelman lähettäminen

Voit lähettää tiedostoja paremmilla sarjaliikennepääteohjelmilla. Tee tekstitiedosto esim excel-taulukolla, jonka sisältö on esimerkiksi

```
#m 100 035  
#m 101 030  
#m 102 071  
#m 103 001  
#m 104 086  
#m 105 097  
#m 106 108  
jne....
```

Pysäytä ohjelmakierto #stop – komennolla ja lähetä tiedosto. (sarjaliikenneohjelma Plazic:ssa vaatii kaikki resurssit ison datamäärän vastaanottamiseen). Käynnistä #start-käskyllä, ja tallenna, jos tarve niin vaatii EPROM:iin #mw-käskyllä.

Digitaalitulolla luetaan kytkimen kiinni/auki tieto ohjelmaan. Digitaalitulo sisältää 30 muistipaikkaa, jolla tarvittavat tiedot syötetään.

Muistipaikat (sisältää vain yhden luvun väliltä 0-255)

- 1-4 Aina yllä olevan kuvan mukaan (Ohjelman sisäinen käyttö)
- 5-12 Tulon nimi (tulostuu näytölle tällä nimellä). Max 8 merkkiä. ASCII taulukon mukaan.  
<http://www.coolbasic.com/cbmanual/ascii.html> (Käytä #rn-käskyä, et tarvitse ASCII taulukkoa)
- 13 Sisältö aina 3. Merkki, joka kertoo, että nimikenttä loppuu tähän.
- 14 Lukuviive 0=sek,1=min. Määritellään viiveen yksikkö. pituus muistipaikassa 24
- 15 Lukuviiveen vaikutus. 0=kummatkin suunnat. 1= vetohidastus(nouseva reuna), 2= päästöhidastus(laskeva reuna)

- 16 0= normaali toiminta. 1=Pehmopiste. Pehmopiste on ohjelmallinen digitaali input. Voit tehdä itse joistakin toiminnoista hälytyspisteen. (Lämpötilaraja, yleishälytys jne). Käyttäytyy samoin kuin dirmaali digitaalitulo, mutta ei ole kytketty fyysiseen kanavaan (liittimeen).
- 17-18 Pehmopisteen osoite. Osoitteessa voi olla tila 0 tai 1. Osoitteen sisältö ohjaa esim hälytyksen päälle. Esimerkiksi voit kytkeä analogiatulon ylärajahälytykseen.
- 19 1=lähetetään tekstiviestillä tilan muutos. 0= ei lähetetä.
- 20-23 Ohjelman sisäinen käyttö (aikalaskuri ym).
- 24 Lukuviive sekuntteina/minuutteina.0-255. esim luku 2 tarkoittaa, että kytkin pitää olla kiinni 2s/2min, ennen kuin tieto lähetetään. . Aika ei ole tarkka!!!
- 25 Näytölle tulostuva hälytysteksti. 0= ei tekstiä. 1= Päällä/Pois. 2 = Hälytys/ Hälytys poistui.
- 26 Ohjelman sisäinen käyttö. (muutoksen tutkiminen)
- 27 Kanava numero. Syötä liittimen numero, mihin olet kytkenyt johtimet.
- 28 0=Suora /1= käänteinen toiminta (invertointi). Tuleeko hälytys, kun kosketin sulkeutuu vai avautuu.
- 29 Kytkimen tila luettavissa tästä muistipaikasta. Jos tieto kytketään toiseen toimilohkoon, lue tästä muistipaikasta.
- 30 Aina 37. Ohjelman sisäinen käyttö. (Ohjelma tietää, että tähän päättyy Digital Input)

## Digitaalilähtö (päälle/pois ohjaus)

Digitaalilähdöllä voi ohjata asioita päälle tai pois. Ohjelman toimilohko sisältää 20 muistipaikkaa, jolla tarvittavat tiedot syötetään.

OUTPUT				
Digitaalilähtö				
Muistipaikka	Sisältö			
100	1	35	#	Tietueen aloitus
101	2	20	20	Tietueen pituus
102	3	100	d	Digitaalilähtö
103	4	1	1	Versio
104	5	76	L	Nimi: Merkki 1
105	6	97	a	Nimi: Merkki 2
106	7	104	h	Nimi: Merkki 3
107	8	116	t	Nimi: Merkki 4
108	9	111	o	Nimi: Merkki 5
109	10	49	1	Nimi: Merkki 6
110	11	3		Nimi: Merkki 7
111	12			Nimi: Merkki 8
112	13	3	3	End of text
113	14			Ei käytössä
114	15	22		Pin numero (kanava)
115	16	0		Invertointi 0=ei, 1=invertoidaan lähtö
116	17	168		Lähtö ohjaus (osoite, mistä luetaan)
117	18	0		
118	19			Lähdön tila
119	20	37	%	Tietueen lopetus

- 1-4 Aina yllä olevan kuvan mukaan (Ohjelman sisäinen käyttö)
- 5-12 Lähdön nimi (tulostuu näytölle tällä nimellä). Max 8 merkkiä. ASCII taulukon mukaan.  
<http://www.coolbasic.com/cbmanual/ascii.html> (Käytä #rn-käskyä, et tarvitse ASCII taulukkoa)
- 13 Sisältö aina ”3”. Merkki, joka kertoo, että nimikenttä loppuu tähän.
- 14 Ei käytössä.
- 15 Kanava numero. Syötä liittimen numero, mihin olet kytkenyt johtimet.
- 16 0=Suora /1= käänteinen toiminta (invertointi). Muuttaa ohjauksen suunnan käänteiseksi.
- 17-18 Lähtö ohjautuu päälle/pois tämän muistipaikan sisällön mukaan. Esimerkiksi kello-ohjauksen lähtö on tässä muistipaikassa. Käytä #ml komentoa!!! (Kaksi muistipaikkaa sisältää yhdessä yhden osoitteen)
- 19 Kytken tila luettavissa tästä muistipaikasta. Jos tieto kytketään toiseen toimilohkoon, lue tästä muistipaikasta.
- 30 Aina 37. Ohjelman sisäinen käyttö. (Ohjelma tietää, että tähän päättyy )

## Analogiatulo (mittaus)

Analogiatuloon voi kytkeä 0-5V portaattomasti muuttuvan jännitteen. Jännite mitataan ja muutetaan (skaalataan) näyttämään jotakin tietoa (paine, paikka, lämpötila jne) Ohjelmassa voi määritellä hälytysrajat. Hälytyksen ylittämisestä saadaan tieto, jolla voidaan ohjata asioita.

ANALOGIA INPUT			
Analogiatulo, Jännitetulo 0-5V			
161	1	35	#
162	2	50	50
163	3	97	a
164	4	1	1
165	5	80	P
166	6	97	a
167	7	105	i
168	8	110	n
169	9	101	e
170	10		
171	11		
172	12		
173	13	3	3
174	14		
175	15		
176	16		
177	17		
178	18		
179	19		
180	20		
181	21		
182	22		
183	23		
184	24		
185	25		
186	26		
187	27		
188	28		
189	29		
190	30		
191	31		
192	32		
193	33		
194	34		
195	35		
196	36		
197	37		
198	38		
199	39		
200	40	b	
201	41	a	
202	42	r	
203	43		
204	44		
205	45		
206	46		
207	47		
208	48		
209	49		
210	50	37	%

= ei tarvitse syöttää mitään

- 1-4 Aina yllä olevan kuvan mukaan (Ohjelman sisäinen käyttö)
- 5-12 Lähdön nimi (tulostuu näytölle tällä nimellä). Max 8 merkkiä. ASCII taulukon mukaan.  
<http://www.coolbasic.com/cbmanual/ascii.html> (Käytä #rn-käskyä, et tarvitse ASCII taulukkoa)
- 13 Sisältö aina ”3”. Merkki, joka kertoo, että nimikenttä loppuu tähän.
- 14-17 Mittaustieto on luettavissa tästä. Voit käyttää tietoa muissa toimilohkoissa
- 18-21 Ylärajahälytys. Asettele raja komennolla #M4 XXX (1. osoite XXX:n tilalle)
- 22-25 Alarajahälytys. Asettele raja komennolla #M4 XXX (1. osoite XXX:n tilalle)
- 26 Muistipaikka on tilassa 1, jos **YLÄ**rajahälytys on ylitetty. Voi käyttää ohjaamaan esim lähtöä.
- 27 Muistipaikka on tilassa 1, jos **AL**arajahälytys on ylitetty. Voi käyttää ohjaamaan esim lähtöä.
- 28-31 Mittausalueen yläraja (skaalaus, mitä mittaus näyttää, kun tulossa on 5V)
- 32-35 Mittausalueen alaraja (skaalaus, mitä mittaus näyttää, kun tulossa on 0V)
- 40-42 Mittausarvon yhteydessä näytettävä yksikkö. ASCII taulukon mukaan.
- 43 1= Tulostaa näytölle mittauksen 1s välein. 0= ei tulosta
- 17-18 Lähtö ohjautuu päälle/pois tämän muistipaikan sisällön mukaan. Esimerkiksi kello-ohjauksen lähtö on tässä muistipaikassa. Käytä #ml komentoa!!! (Kaksi muistipaikkaa sisältää yhdessä yhden osoitteen)
- 47 Kanava numero. Syötä liittimen numero, mihin olet kytkenyt johtimet.
- 48 Suodatus. Keskiarvoistaa näytettävän mittauksen. 0-255 (x10 vanhaa mittausta ka)
- 30 Aina 37. Ohjelman sisäinen käyttö. (Ohjelma tietää, että tähän päättyy Digital Input)



# Termistoritulo (lämpötilamittaus)

Termistor input			
Vastuslämpöanturi (NTC 1000 Ohm+1kOhm sarjavastus)			
Muistipaikka			
0	1	35	#
1	2	50	50
2	3	116	t
3	4	1	1
4	5	80	P
5	6	97	a
6	7	105	i
7	8	110	n
8	9	101	e
9	10		
10	11		
11	12		
12	13	3	3
13	14		
14	15		
15	16		
16	17		
17	18		
18	19		
19	20		
20	21		
21	22		
22	23		
23	24		
24	25		
25	26		
26	27		
27	28		
28	29		
29	30		
30	31		
31	32		
32	33		
33	34		
34	35		
35	36		
36	37		
37	38		
38	39		
39	40		b
40	41		a
41	42		r
42	43	0	
43	44		
44	45		
45	46	1	
46	47		
47	48	10	
48	49		
49	50	37	%
Tietueen aloitus			
Tietueen pituus			
Termistor input			
Versio			
Nimi: Merkki 1			
Nimi: Merkki 2			
Nimi: Merkki 3			
Nimi: Merkki 4			
Nimi: Merkki 5			
Nimi: Merkki 6			
Nimi: Merkki 7			
Nimi: Merkki 8			
End of text			
MITTAUS			
Ylärajahälytys			
Alarajahälytys			
Ylärajahälytys päällä DO			
Alarajahälytys päällä DO			
Offset =0 (#m4 komento) (kalibrointi)			
Slope =1 (#m4 komento) (kalibrointi)			
Ei käytössä			
Yksikkö			
Yksikkö			
Yksikkö			
1= tulostaa näytölle mittauksen 1s välein			
Hälytys tulostus 0=ei, 1=päällä/pois			
Pin numero (kanava)			
Suodatus. 1-255 (x10 vanhaa mittausta ka) (Eikäytössä)			
Tietueen lopetus			
= ei tarvitse syöttää mitään			
Käytä #m4 / #p4 käskyjä (lukuluku käyttää neljä muistipaikkaa)			
Kanava / Liitin, mihin olet kytkenyt johdot			
Käytä #m1 komentoa			

## OR/NOR/NOT (Käytä yhtä tuloa)

131	1	35	#	Tietueen aloitus
132	2	30	30	Tietueen pituus
133	3	79	O	OR Tietueen nimi
134	4	1	1	Versio
135	5	80	P	Nimi: Merkki 1
136	6	97	a	Nimi: Merkki 2
137	7	115	s	Nimi: Merkki 3
138	8	105	i	Nimi: Merkki 4
139	9	3		Nimi: Merkki 5
140	10			Nimi: Merkki 6
141	11			Nimi: Merkki 7
142	12			Nimi: Merkki 8
143	13	3	3	End of text
144	14	2	2	Tulojen lukumäärä 2-4
145	15		295	Tulo 1 Osoite (Low byte)
146	16			(High Byte)
147	17		297	Tulo 2 Osoite (Low byte)
148	18			(High Byte)
149	19			Tulo 3 Osoite (Low byte)
150	20			(High Byte)
151	21			Tulo 4 Osoite (Low byte)
152	22			(High Byte)
153	23			Lähdön invertointi (NOR)
154	24			Varaus
155	25			Varaus
156	26			Varaus
157	27			Varaus
158	28			Varaus
159	29			Lähtö 0/1 (kaikki yli 1 on 1)
160	30	%	37	Tietueen lopetus

Huom 3-4 inputs ei vielä koodissa!!!

Invertointi ei vielä pelaa.

HUOM älä jätä tulojen osoitteeksi osoitetta 0. Siinä osoitteessa on "roskaa".

	<b>RS</b>			
	Muistitoiminto			
150	1	#	35	Tietueen aloitus
151	2	20	20	Tietueen pituus
152	3	R	82	RS Toimminto
153	4	1	1	Versio
154	5			Nimi: Merkki 1
155	6			Nimi: Merkki 2
156	7			Nimi: Merkki 3
157	8			Nimi: Merkki 4
158	9			Nimi: Merkki 5
159	10			Nimi: Merkki 6
160	11			Nimi: Merkki 7
161	12			Nimi: Merkki 8
162	13			End of text
163	14	10		SET Osoite (Low byte)
164	15			(High Byte)
165	16	11		RESET Osoite (Low byte)
166	17			(High Byte)
167	18			
168	19			Lähtö 0 / 1 (kaikki yli 1 on 1)
169	20	¼	37	Tietueen lopetus

	GSM numero			
320	1	#	35	Tietueen aloitus
321	2	30	30	Tietueen pituus
322	3	110	n	GSM Numeron talletus
323	4	1	1	Versio
324	5		+	Puhelinnumero (max 20 merkkiä)
325	6		3	
326	7		5	
327	8		8	
328	9		4	
329	10		0	
330	11		1	
331	12		9	
332	13		5	
333	14		5	
334	15		3	
335	16		3	
336	17		7	
337	18	3	3	End of text
338	19			
339	20			
340	21			
341	22			
342	23			
343	24			
344	25			
345	26			
346	27			SET Puh.Nro aktiivinen(Low byte) (VARAUS)
347	28			(High Byte)
348	29			Aktiivinen numero 1
349	30	٪	37	Tietueen lopetus

MP 27 ei käytössä vielä.

GSM Ohjaus				
Vertaa tekstiviestin sisältöä ja jos sama -> lähtö käyväisee tilassa "1"				
100	1	#	35	Tietueen aloitus
101	2	30	30	Tietueen pituus
102	3	G	71	GSM Ohjaus
103	4	1	1	Versio
104	5	118	v	Nimi: Merkki 1
105	6	97	a	Nimi: Merkki 2
106	7	108	l	Nimi: Merkki 3
107	8	111	o	Nimi: Merkki 4
108	9	3	3	End of text
109	10			
110	11			
111	12			
112	13			
113	14			
114	15			
115	16			
116	17			
117	18			
118	19			
119	20	3	3	End of text
120	21			
121	22			
122	23			
123	24			
124	25			
125	26			
126	27			
127	28			
128	29			Lähtö 0 / 1 (kaikki yli 1 on 1)
129	30	٪	37	Tietueen lopetus

GSM ohjaus ”kuuntelee” tekstiviestejä. Jos muistipaikkojen 104-> sisältö vastaa viestissä esiintyvää merkkijonoa, asettuu lähtö hetkeksi tilaan 1 (yksi ohjelmakierto). Käytä RS-toimilohkoa ohjauksessa.

## KELLO-OHJAUS

Muistipaikka	Sisältö	
150	1	35 # Tietueen aloitus
151	2	20 20 Tietueen pituus
152	3	67 C Kello-ohjaus
153	4	1 1 Versio
154	5	76 L Nimi: Merkki 1
155	6	97 a Nimi: Merkki 2
156	7	104 h Nimi: Merkki 3
157	8	116 t Nimi: Merkki 4
158	9	111 o Nimi: Merkki 5
159	10	49 1 Nimi: Merkki 6
160	11	3 Nimi: Merkki 7
161	12	Nimi: Merkki 8
162	13	3 3 End of text
163	14	Päälle klo: Tunnit
164	15	Päälle klo: Minuutit
165	16	Pois klo: Tunnit
166	17	Pois klo: Minuutit
167	18	Varaus
168	19	Lähdön tila
169	20	37 % Tietueen lopetus

Kytke Digitaali lähtö paikkaan 19, niin saat kello-ohjauksen tehtyä.

Laita muistipaikkaan 14 osoite millä ohjaat servoa. Esim jos osoite 12 on vapaa, kirjoita #ml xxx 012. (xxx on rivin 14 osoite, kuvan esimerkissä 1012).

Voit ohjata servoa kirjoittamalla muistipaikkaan 12 lukuarvon 0-255.

Uusia toimilohkoja Beta 2 versiosta alkaen:

### **KOMPARAATTORI**

Vertailee, onko tuleva tieto isompi tai pienempi, kuin aseteltu raja. Jos raja on ylitetty, lähtö asettuu tilaan "1". (osoiterivit 31 yläraja, 32 alaraja ja 39 yhteishälytys)

Samanaikaisesti pakotetaan lähtö (osoiterivi 35) olemaan aina min-max välissä.

	Komparaattori / Limitteri									
	Vertailee, onko mittaus suurempi/pienempi, ja ohjaa lähdön päälle									
430	1	35	#	Tietueen aloitus						
431	2	40	40	Tietueen pituus						
432	3	76	L							
433	4	1	1	Versio						
434	5			Nimi: Merkki 1						
435	6			Nimi: Merkki 2						
436	7			Nimi: Merkki 3						
437	8			Nimi: Merkki 4						
438	9			Nimi: Merkki 5						
439	10			Nimi: Merkki 6						
440	11			Nimi: Merkki 7						
441	12			Nimi: Merkki 8						
442	13	3	3	End of text						
443	14			Ei käytössä						
444	15	1	1	0=vakio Float type / 1=osoite	Luku 1.			Osoite, mistä data luetaan		
445	16			Vakio 1. tavu / Osoite 1. tavu						
446	17			Vakio 2. tavu / Osoite 2. tavu						
447	18			Vakio 3. tavu						
448	19			Vakio 4. tavu						
449	20			0=vakio Float type / 1=osoite	Luku 2.			Yläraja		
450	21			Vakio 1. tavu / Osoite 1. tavu						
451	22			Vakio 2. tavu / Osoite 2. tavu						
452	23			Vakio 3. tavu						
453	24			Vakio 4. tavu						
454	25			0=vakio Float type / 1=osoite	Luku 3.			Alaraja		
455	26			Vakio 1. tavu / Osoite 1. tavu						
456	27			Vakio 2. tavu / Osoite 2. tavu						
457	28			Vakio 3. tavu						
458	29			Vakio 4. tavu						
459	30			Ei käytössä						
460	31			Yläraja ylitetty, DIGITAL OUTPUT				Digitaali lähtö ohjaukseen		
461	32			Alaraja alitettu, DIGITAL OUTPUT				Digitaali lähtö ohjaukseen		
462	33									
463	34									
464	35			Lähtö (rajoitettu min - max väliin)	Lähtö					
465	36									
466	37									
467	38									
468	39			Hälytys päällä (min TAI max)				Digitaali lähtö ohjaukseen		
469	40	37	%	Tietueen lopetus						
				= ei tarvitse syöttää mitään						
				Käytä #m4 / #p4 käskyjä (liukuluku käyttää neljä muistipaikkaa)						
				Kanava / Liitin, mihin olet kytkenyt johdot						
				Käytä #ml kommentoa						

## PI- Säädin

Säädin säätää mittauksen ja asetusarvon eron perusteella säätimen lähtöä.

Muistipaikka	Rivi				
0	1	35	#	Tietueen aloitus	
1	2	70		Tietueen pituus	
2	3	80	P	PI Controller	
3	4	1	1	Versio	
4	5	80	P	Nimi: Merkki 1	
5	6	97	a	Nimi: Merkki 2	
6	7	105	i	Nimi: Merkki 3	
7	8	110	n	Nimi: Merkki 4	
8	9	101	e	Nimi: Merkki 5	
9	10			Nimi: Merkki 6	
10	11			Nimi: Merkki 7	
11	12			Nimi: Merkki 8	
12	13	3	3	End of text	
13	14			Measure address	Osoite, mistä säätimen mittausta luetaan
14	15			Ei käytössä	
15	16				
16	17			Setpoint	Asetusarvo säätimelle
17	18				
18	19				
19	20				
20	21				
21	22			Range MIN	Mittauksen alaraja
22	23				
23	24				
24	25				
25	26			Range MAX	Mittauksen yläaraja
26	27				
27	28				
28	29				
29	30			Deviation (Ero arvo)	
30	31				
31	32				
32	33				
33	34	10		P (0-255 -> 0...25.5)	Säätinparametrit
34	35	60		TI (0-255 -> 0 ...255s)	
35	36			Td (0-255 -> 0...25.5s) EI KÄYTÖSSÄ	
36	37			AUTO/MAN (0=MAN, 1 = AUTO)	Automaatti / Käsiajo
37	38			P-vaikutus	
38	39			(Säätimen sisäinen käsittely)	
39	40				
40	41				
41	42			I-vaikutus	
42	43			(Säätimen sisäinen käsittely)	
43	44				
44	45				
45	46			D-vaikutus	
46	47			(Säätimen sisäinen käsittely)	
47	48				
48	49				
49	50			Controller output	Huom laita nollat!!!, muuten lähtö nan"
50	51				
51	52			Lue tästä säätimen lähtö	
52	53				
53	54		b	Yksikkö	
54	55		a	Yksikkö	
55	56		r	Yksikkö	
56	57			Output direction: 0=normal, 1=reverse	Ei käytössä
57	58				
58	59				
59	60				
60	61				
61	62				
62	63				
63	64				
64	65			TULOSTUS 0=ei, 1=päällä/pois	
65	66			Hälytys tulostus 0=ei, 1=päällä/pois	Ei käytössä
66	67	100		Lähtö rajoitus max 0-100	
67	68	0		Lähtö rajoitus min 0-100	
68	69			Lähtö 0-180 (RC servolle)	
69	70	37	%	Tietueen lopetus	

= ei tarvitse syöttää mitään

Käytä #m4 / #p4 käskyjä (liukuluku käyttää neljä muistipaikkaa)

Kanava / Liitin, mihin olet kytkenyt johdot

Käytä #ml komentona



## Write Float type variable

(edge triggered, + and - function included)

320	1	35	#	Tietueen aloitus
321	2	40	40	Tietueen pituus
322	3	87	W	WRITE F Tietueen nimi
323	4	1	1	Versio
324	5			Nimi: Merkki 1
325	6			Nimi: Merkki 2
326	7			Nimi: Merkki 3
327	8			Nimi: Merkki 4
328	9			Nimi: Merkki 5
329	10			Nimi: Merkki 6
330	11			Nimi: Merkki 7
331	12			Nimi: Merkki 8
332	13	3	3	End of text
333	14			
334	15			Active bit Osoite (Low byte)
335	16			Edge triggered
336	17			Float Osoite (Low byte)
337	18			(High Byte)
338	19		0	Vakio, mikä lisätään edelliseen
339	20			
340	21			
341	22			
342	23	0		Lähdön Osoite (Low byte)
343	24	0		(osoite 0 -> lähtöä ei kirjoiteta)
344	25			Lähtö
345	26			
346	27			
347	28			
348	29			
349	30			
350	31			
351	32			
352	33			
353	34			
354	35			
355	36			
356	37			
357	38			
358	39			Lähtö 0 / 1 (kaikki yli 1 on 1)
359	40	37	%	Tietueen lopetus

## Huom!!!!

Oltava tarkkana lohkon käytössä.

Ainut lohko (Write Byten:n lisäksi), jolla voi kirjoittaa väärälle muistialueelle!!!

Luku 1 Tämä luku luetaan osoitteesta

Luku 2 Tämä luku on vakio

Summa Yhteenlaskettu summa

= ei tarvitse syöttää mitään

Käytä #m4 / #p4 käskyjä (liukuluku käyttää neljä muistipaikkaa)

Kanava / Liitin, mihin olet kytkenyt johdot

Käytä #ml kommentoa

## Write Byte (1-4 byte)

(kirjoittaa tavun/tavut muistipaikkaan)

140	1	35	#	Tietueen aloitus
141	2	40	40	Tietueen pituus
142	3	87	W	WRITE F Tietueen nimi
143	4	2	2	Versio
144	5			Nimi: Merkki 1
145	6			Nimi: Merkki 2
146	7			Nimi: Merkki 3
147	8			Nimi: Merkki 4
148	9			Nimi: Merkki 5
149	10			Nimi: Merkki 6
150	11			Nimi: Merkki 7
151	12			Nimi: Merkki 8
152	13	3	3	End of text
153	14	0	0	0=reunaliipaisu, 1= jatkuva kirjoitus
154	15			<b>Active bit</b> Osoite (Low byte)
155	16			Edge triggered
156	17			Osoite 1 (low byte)
157	18			(high byte)
158	19	0		Osoite 2 (low byte)
159	20	0		(high byte)
160	21	0		Osoite 3 (low byte)
161	22	0		(high byte)
162	23	0		Osoite 4 (low byte)
163	24	0		(high byte)
164	25			1. Data low
165	26			2. Data low
166	27			3. Data low
167	28			4. Data low
168	29			1. Data high
169	30			2. Data high
170	31			3. Data high
171	32			4. Data high
172	33			Varaus
173	34			
174	35			
175	36			
176	37			
177	38			
178	39			Lähtö 0 / 1 (kaikki yli 1 on 1)
179	40	37	%	Tietueen lopetus

## Huom!!!!

Oltava tarkkana lohkon käytössä.

Ainut lohko (Write Float:n lisäksi), jolla voi kirjoittaa väärälle muistialueelle!!!

Ohje: syötä kaikki rivit huolella. Lopuksi tietueen loppumerkki 37, niin ohjelmalohko käynnistyy.

Reunaliipaisu: Nouseva reuna= data high, Laskeva reuna Data low

Osoite, mihin toimilohko kirjoittaa!!!

Jos osoite 0, dataa ei kirjoiteta

Jos osoite 0, dataa ei kirjoiteta

Jos osoite 0, dataa ei kirjoiteta

= ei tarvitse syöttää mitään

Käytä #m4 / #p4 käskyjä (liukuluku käyttää neljä muistipaikkaa)

Kanava / Liitin, mihin olet kytkenyt johdot

Käytä #ml komentoa

[illegible]

Ottaa liukuluvusta tietyn välin, ja skaalaa sen lähdeksi. Käyttö esim toimilaitteportaissa.

				Tietueen aloitus			Tulo esimerkiksi 0....100%
520	1	#	35	Tietueen pituus			Alaraja 50, Yläraja 100.
521	2	40	40	XXXXX			Lähdön Skaala 100%
522	3	115	s	Versio			-> kun tulo 0-50, lähtö=0
523	4	1	1	Nimi: Merkki 1			Tulo 50-100, lähtö 0...100
524	5			Nimi: Merkki 2			
525	6			Nimi: Merkki 3			
526	7			Nimi: Merkki 4			
527	8			Nimi: Merkki 5			
528	9			Nimi: Merkki 6			
529	10			Nimi: Merkki 7			
530	11			Nimi: Merkki 8			
531	12			End of text			
532	13	3	3	Tulostus =1			
533	14			0=vakio Float type / 1=osoite			
534	15	1	1	Vakio 1. tavu / Osoite 1. tavu	Luku 1.		Tulo
535	16			Vakio 2. tavu / Osoite 2. tavu			
536	17						
537	18						
538	19						
539	20		0	0=vakio Float type / 1=osoite	Luku 2.		Yläraja
540	21			Vakio 1. tavu / Osoite 1. tavu			
541	22			Vakio 2. tavu / Osoite 2. tavu			
542	23			Vakio 3. tavu			
543	24			Vakio 4. tavu			
544	25		0	0=vakio Float type / 1=osoite	Luku 3.		Alaraja
545	26			Vakio 1. tavu / Osoite 1. tavu			
546	27			Vakio 2. tavu / Osoite 2. tavu			
547	28			Vakio 3. tavu			
548	29			Vakio 4. tavu			
549	30		0	0=vakio Float type / 1=osoite	Luku 4.		Lähdön skaala 0.... Tämä luku
550	31			Vakio 1. tavu / Osoite 1. tavu			
551	32			Vakio 2. tavu / Osoite 2. tavu			
552	33			Vakio 3. tavu			
553	34			Vakio 4. tavu			
554	35			Lähtö	Lähtö		
555	36						
556	37						
557	38						
558	39			Lähtö 0...255			
559	40	%	37	Tietueen lopetus			
				= ei tarvitse syöttää mitään			



// Versio Beta b01\_80  
// Suomi - Finland - Jämsänkoski  
//  
// (Ohjelman tekijä ei ota mitään vastuuta ohjelman aiheuttamista aineellisista vahingoista)  
//  
// Ohjelmaa saa käyttää omaan yksityiseen käyttöön ilman rahallista korvausta. Jos tuotetta haluaa käyttää kaupallisesti (myy tuotetta, asentaa ohjelmiston tuotteeseen  
// jota myydään, hyödyntää ohjelman perusrakennetta kopioimalla tai muokkaamalla alkuperäistä), on sovittava kirjallisesti ohjelman omistajan kanssa (plazic.owner@outlook.com).  
// Ohjelmasta ei saa poistaa näitä otsikko tai käyttöoikeusrivejä.  
//  
//  
// Ohjelman muokkaaminen: Ohjelmaa saa muokata, parantaa, kehittää vapaasti, kunhan ei muuta edellä mainittuja käyttöoikeuksia. Ohjelmasta olisi hyvä saada  
// kopio, ohjelmistoa voitaisiin kehittää eteenpäin.  
// Ohjelman oppilaitoskäyttö on sallittua

Ohjelman kaupallisesta hyödyntämisestä voit sopia [plazic.owner@outlook.com](mailto:plazic.owner@outlook.com)

[/ Pasi Kuusela](#)